(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE USE OF WINDOW FUNCTIONS AND
KALMAN FILTERING IN SPECTRAL ESTIMATION

by

William W. Go

March 1990

Thesis Advisor: Ralph Hippenstiel

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) EC	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) THE USE OF WINDOW FUNCTIONS AND KALMAN FILTERING IN SPECTRAL ESTIMATION

12 PERSONAL AUTHOR(S)
GO, William W.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's thesis	FROM _____ TO _____	1990 March	137

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense and the US Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	signal processing; Kalman filter; periodogram; spectral analysis

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The periodogram, the square of the magnitude of the Fourier Transform, is widely used to estimate the spectral content of sampled processes. The performance of the periodogram is degraded by spectral leakage. This is the consequence of processing finite-length data records. Classical means of enhancing periodogram performance are the use of tapered window functions and averaging of several periodograms. These methods smooth the spectral estimate, but at a loss of resolution. A non-stationary Kalman filter was applied to the periodogram of unwindowed (i.e., rectangular windowed) time data in an effort to smooth the noise portions of the periodogram while leaving the main spectral response unaltered. The Kalman filter was able to enhance the periodogram. Best results were obtained in the single spectral peak case. Even in the case of multiple spectral peaks, the resolution of the unfiltered periodogram was largely preserved

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL HIPPENSTIEL, R.	22b TELEPHONE (Include Area Code) 408-646-2633 / 22c OFFICE SYMBOL EC/Hi

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE
S/N 0102-LF-014-6603 UNCLASSIFIED

19. cont.
since the filtering algorithm was designed to selectively smooth the noise-only segments of the spectral estimate.

Approved for public release; distribution is unlimited

The Use of Window Functions and Kalman Filtering in Spectral Estimation

by

William W. Go
Captain, United States Marine Corps
B.A., University of Pennsylvania, 1980

Submitted in partial fulfillment of the requirements of degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1990

Author: _____
William W. Go

Approved by: _____
Ralph Hippenstiel, Thesis Advisor

_____
Roberto Cristi, Second Reader

_____
John P. Powers, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

The periodogram, the square of the magnitude of the Fourier Transform, is widely used to estimate the spectral content of sampled processes. The performance of the periodogram is degraded by spectral leakage. This is the consequence of processing finite-length data records. Classical means of enhancing periodogram performance are the use of tapered window functions and averaging of several periodograms. These methods smooth the spectral estimate, but at a loss of resolution. A non-stationary Kalman filter was applied to the periodogram of untapered (i.e., rectangular windowed) time data in an effort to smooth the noise portions of the periodogram while leaving the main spectral response unaltered. The Kalman filter was able to enhance the periodogram. Best results were obtained in the single spectral peak case. Even in the case of multiple spectral peaks, the resolution of the unfiltered periodogram was largely preserved since the filtering algorithm was designed to selectively smooth the noise-only segments of the spectral estimate.

TABLE OF CONTENTS

I. INTRODUCTION

The periodogram, the square of the magnitude of the Fourier Transform, is widely used to estimate the spectral content of sampled processes. The periodogram remains popular in the face of more modern spectral estimation techniques (i.e., parametric modeling) due to its low cost and ease of implementation in real time. The performance (ability to detect signals in noise) of the periodogram is degraded by window function sidelobe effects. This is the unavoidable consequence of processing data records of finite length. In addition, the periodogram may have a fairly large variance (i.e., mean equals the standard deviation under noise-only conditions). A classical means of enhancing the performance of the periodogram is the use of tapered window functions, such as the Hamming window, in order to minimize the effects of the discontinuity at the boundaries of the finite observation. Another common method is to average a series of periodograms in an effort to smooth the spectral estimate (i.e., reduce the variance of the estimate). Almost invariably, the consequences of these techniques are a broadening of the main spectral peaks and a corresponding loss of spectral resolution. What is proposed here is an application of a non-stationary Kalman filter to the sequence presented by the periodogram of untapered (i.e., rectangular

windowed) time data. The objective is to filter (smooth) the noise portions of the spectral estimate and leave the main spectral responses unaltered. The result is that the dominant spectral peaks will be highlighted against the noise "floor" out of which they rise. Since the main spectral peaks are unaltered, the resolution of the original periodogram is preserved. Using the test cases of single and multiple sinusoids in Gaussian white noise, the Kalman filter's performance was evaluated for signal detectability and resolution at different input signal-to-noise ratios on multiple noise realizations. The effects of varying the filter's detection parameter and the data/transform length were also investigated.

II. CLASSICAL SPECTRAL ESTIMATION

A. BACKGROUND

Estimation of the power spectral density (PSD) of sampled deterministic or stochastic processes is usually based on techniques employing the Fast Fourier Transform (FFT). These techniques are computationally efficient and produce good results for many different types of signals. There are, however, two significant limitations associated with the FFT-based techniques. First and foremost is the problem of frequency resolution, that is, the ability to distinguish between the presence of one or several spectral components in a given sample set of data. Frequency resolution of stationary signals varies with the specific technique employed but, in general, it is proportional to the reciprocal of the time interval represented by the sample. The second limitation of the FFT-based methods is caused by the windowing of the data that occurs during processing. Windowing causes "leakage" in the spectral domain. Energy in the main lobe of a spectral response "leaks" into adjacent sidelobes, obscuring and distorting the spectral responses due to other frequency components that may be present. In some cases, weak spectral responses may be completely masked by the sidelobes of stronger spectral responses and thus go undetected. Careful selection and use of tapered data windows can reduce sidelobe

leakage, but always at the cost of reduced frequency resolution [Ref. 1].

B. CLASSICAL SPECTRAL ESTIMATION TECHNIQUES

The two best-known classical spectral estimation techniques are the Blackman-Tukey method and the periodogram. The Blackman-Tukey approach, introduced in 1958 [Ref. 2], first estimates the autocorrelation function from the data and then Fourier transforms the correlation estimates to obtain a power spectral density estimate. The Blackman-Tukey spectral estimator is given by:

$$\hat{P}_{BT}(f) = \sum_{k=-(N-1)}^{N-1} \hat{r}_{xx}(k)\exp(-j2\pi fk) \qquad (2.1)$$

where

$$\hat{r}_{xx}(k) = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-1-k} x^*(n)x(n+k); & k = 0,1,2,\ldots,(N-1) \\ \hat{r}_{xx}^*(-k); & k = -(N-1),-(N-2),\ldots,-1 \end{cases} . \qquad (2.2)$$

This is a biased estimator of the true autocorrelation function since:

$$E[\hat{r}_{xx}(k)] = \frac{N-|k|}{N} r_{xx}(k); \quad |k| \le (N-1) . \qquad (2.3)$$

The mean value of the autocorrelation function estimator shows that a triangular (Bartlett) window is applied to the

true autocorrelation function. It is possible to use an unbiased autocorrelation function estimator by replacing the normalization by 1/N in (2.2) with 1/(N-|k|). This, however, can lead to a negative spectral estimate since the unbiased autocorrelation estimator does not guarantee a positive semi-definite sequence. The Blackman-Tukey approach was the most popular spectral estimation technique until the introduction of the FFT algorithm [Refs. 3 and 4].

The periodogram spectral estimate is obtained from the square of the magnitude of the Fourier transform of the data. The data may be weighted by a window function and/or zero-padded. The true spectral estimator is given by:

$$P_{xx}(f) = \lim_{M\to\infty} E\left[\frac{1}{2M+1}\left|\sum_{n=-M}^{M} x(n)\exp(-j2\pi fn)\right|^2\right] \quad . \qquad (2.4)$$

If we ignore the expectation operator and use only the available data, the spectral estimator, denoted as the periodogram, is given by:

$$P_{PER}(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)\exp(-j2\pi fn)\right|^2 \quad . \qquad (2.5)$$

The periodogram produces best results when an integer multiple of periods of constituent frequency components are present in the observation. Despite the advent of more modern

techniques, the periodogram remains a popular means of spectral estimation because it can be easily and inexpensively implemented in real time.

In general, the Blackman-Tukey and the periodogram spectral estimates are not identical. If, however, the biased autocorrelation estimate (2.2) is used and as many autocorrelation lags as data samples (N) are computed, then the Blackman-Tukey and periodogram estimators yield identical numerical results.

C. WINDOW FUNCTIONS

Every set of data is finite in duration. Processing a finite duration observation presents special problems to the harmonic analysis of the data. Some considerations should be given to detectability of spectral components in the presence of nearby strong components and their resolvability. Let the data to be processed consist of N uniformly-spaced samples of the observed signal. The FFT, the basis of the periodogram spectral estimator, assumes sequences to be periodic. In other words, the sample set under analysis is assumed to be one complete period of an infinitely long periodic sequence. The selection of a finite time interval of NT seconds, where T is the time between samples, and of the orthogonal trigonometric basis over this interval leads to an interesting peculiarity of the spectral expansion. From the continuum of possible frequencies, only those which coincide with the basis

functions (the bin centers of the FFT) will project onto a single basis vector. All other frequencies will exhibit non-zero projections on the entire basis set. This phenomena is called spectral leakage and is a consequence of processing finite duration data records [Ref. 1].

Spectral components with frequencies other than those corresponding to the FFT bin centers will typically be present in the observed data. Components with frequencies not at bin centers are not periodic in the observation window. The periodic extension of a signal which does not coincide with the natural periods of its constituent frequency components exhibits discontinuities at the boundaries of the observation. These discontinuities are responsible for spectral contributions (leakage) over the entire range of the FFT frequency bins.

Since we are constrained to deal only with finite-length data, we are forced to make certain assumptions about the data outside of the observation interval. The finite data record may be considered as having been obtained by multiplying an infinite length data sequence with a simple rectangular function:

$$w(n) = \begin{cases} 1; & n = 0,1,2,\ldots,(N-1) \\ 0; & \text{otherwise} \end{cases} \qquad (2.6)$$

The assumption that the data outside of the observation window is zero is unrealistic but unavoidable. Thus, data taken "as is" is actually rectangularly windowed. Non-rectangular window functions are weighting functions applied to the received data in order to reduce the spectral leakage associated with finite observation intervals. The purpose of the window is to reduce the magnitude of the discontinuity at the boundaries of the periodic extension. The goal of windowing is, therefore, to smoothly taper the data record at the boundaries.

By the Convolution Theorem, multiplication of the time series by a window function corresponds in the frequency domain to the convolution of the transforms of the signal sequence and the window function. If we are using a rectangular window and attempting to detect a narrow-band signal, such as a sinusoid in noise, and the sinusoidal frequency is not at a bin center, the convolution will spread or smear some signal power into adjacent frequencies. Conversely, if the sinusoid is at a bin center, then we will see only the zero crossings of the window transform, and experience no leakage. If we are using a non-rectangular window (i.e., a Hamming window), the convolution operation will smear the signal power into adjacent frequencies regardless of the sinusoidal frequency being at a bin center or not.

Leakage has an obvious negative effect on the detection and estimation of sinusoidal components. Sidelobes from adjacent frequency components may add in an unpredictable fashion to the spectral peak of a weak signal, thus distorting the power estimate of that signal. In extreme cases, the sidelobes of strong frequency components may completely mask the main lobe of nearby weaker signals [Ref. 3].

In general, the convolution of the window transform with the signal transform means that the main lobe width of the window transform is the limiting factor (in terms of spectral response) that allows separation of two closely-spaced spectral lines. For a rectangular window, the main lobe width between the 3-dB levels of the resulting digital sinc function (the FFT of a rectangle function) is approximately the reciprocal of the observation interval NT. Leakage effects can be reduced by the use of windows with non-uniform weighting, such as the Blackman and Hamming windows.

Consider, for example, the problem of detecting a sinusoidal signal embedded in Gaussian white noise. Assuming that the observation interval does not contain an integer multiple of periods of the sinusoid, then the frequency of the sinusoid is not at a bin center of the FFT. Some spectral leakage will occur. Recall from basic Fourier theory that the transform of a sinusoid (say a cosine function) is a pair of delta functions given by:

$$\cos(2\pi f_0 t) \xrightarrow{\text{F.T.}} \pi\left(\delta(2\pi f - 2\pi f_0) + \delta(2\pi f + 2\pi f_0)\right) \qquad (2.7)$$

Assuming that the data is obtained by rectangular windowing of an infinitely long sequence (i.e., multiplication of the time series by the window function), then the periodogram will be, by the Convolution Theorem, the square of the magnitude of the convolution of the delta function pair with the Discrete Fourier Transform of the rectangle function (a digital sinc function). The digital sinc function is of the form:

$$D_N(f) = T\exp(-j2\pi fT[N-1])\frac{\sin(\pi fNT)}{\sin(\pi fT)} \quad . \qquad (2.8)$$

Recall from Fourier theory that the convolution of some function, call it F(f), with a delta function, results in the translation of F(f) to the location of the delta function. In this case, the sinc function will be shifted to the location of the delta function dictated by the signal frequency. If the location of the delta function does not exactly coincide with a bin center of the FFT, leakage will occur.

At this point, some discussion of zero-padding is in order. Zero-padding the data sequence prior to the Fourier transformation will not improve the resolution of the periodogram. The purpose of zero-padding is twofold. First, it will interpolate additional power spectral density values in the interval $[-f_s/2, f_s/2]$, where f_s is the sampling frequency [Ref. 3] between those that would have been obtained in a non-zero-padded transform. Second, since the number of

observed data points is not always a power of two, zero-padding is necessary to make the sequence length a power of two to allow the use of a FFT. Consider the Discrete Fourier Transform of an eight-point rectangular window. We know that this transform will produce a digital sinc function. However, when we actually compute and plot the transform, we observe only a central spike at the zero spectral location. (Figure 1). Why do we not see any of the side lobe structure that we know must be present? The side lobes are in fact there. They are not visible because the FFT of the non-zero-padded time series interrogates the resultant digital sinc function at its zero-crossings and hence, the side lobe structure is invisible to us. In other words, the FFT bin centers are coincident with the digital sinc's zero-crossings. Now examine what happens when the eight-point rectangle is zero-padded to sixteen points and then transformed (Figure 2). The side lobes are now clearly visible because we are interpolating a point in between the bin centers of the previous eight-point (non-zero-padded) transform. This principle can now be extended to an actual spectral estimation example.

Consider a unit amplitude sinusoid embedded in Gaussian white noise. In this example, the number of data points N is 64 and a rectangular window is used. The sinusoidal frequency is 10.0 Hz and the sampling frequency, f_s, is 64.0 Hz. The variance of the additive Gaussian white noise is 1/2000. This

corresponds to a signal-to-noise ratio (SNR) of 30 dB where SNR is defined as:

$$SNR = 10\log_{10}\left[\frac{\left(\frac{\text{sinusoidal amplitude}}{2}\right)}{\left(\begin{matrix}\text{noise}\\ \text{variance}\end{matrix}\right)}\right] = 10\log_{10}\left[\frac{\left(\frac{A}{2}\right)}{\left(\sigma^2\right)}\right] , \qquad (2.9)$$

where A = amplitude of the sinusoid.

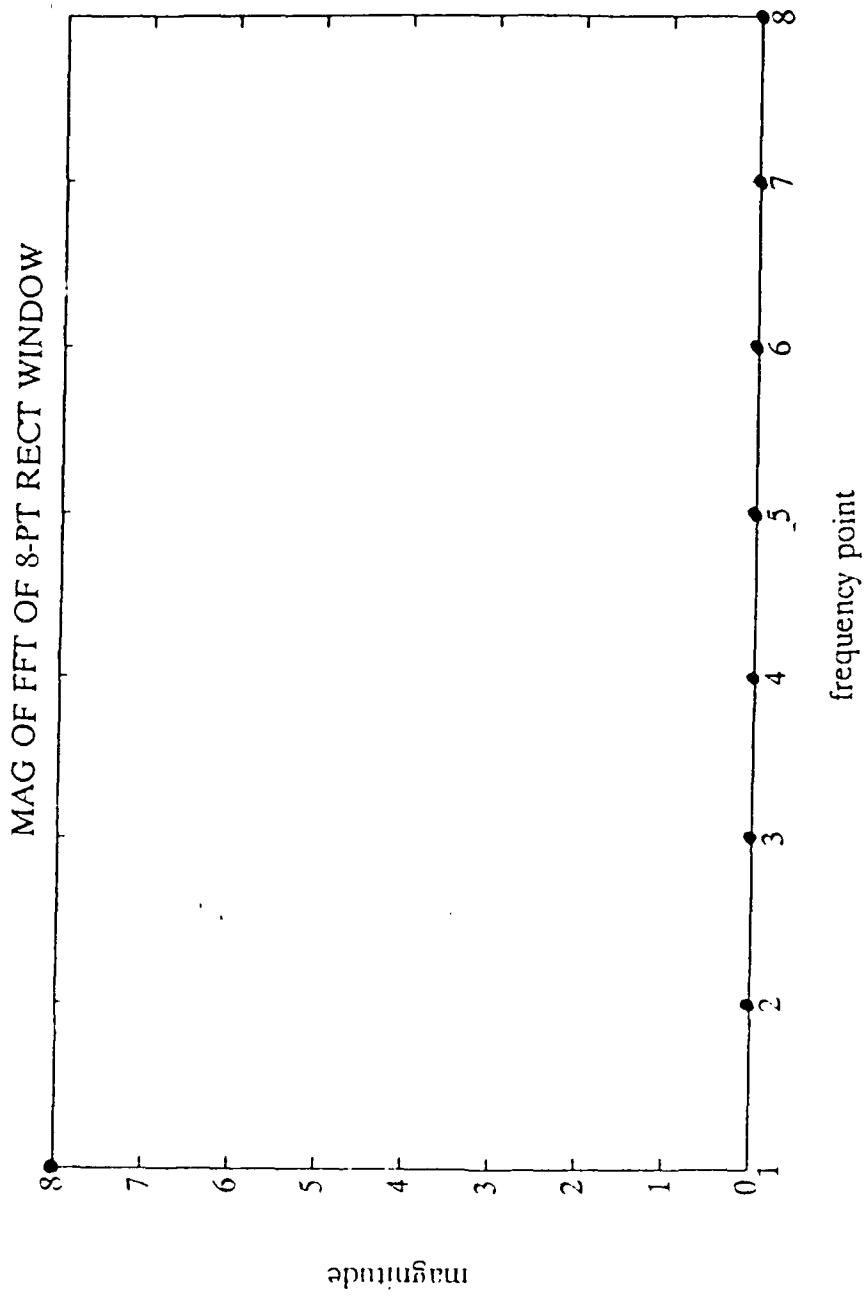


Figure 1. Magnitude of FFT of 8-point Rectangular Window

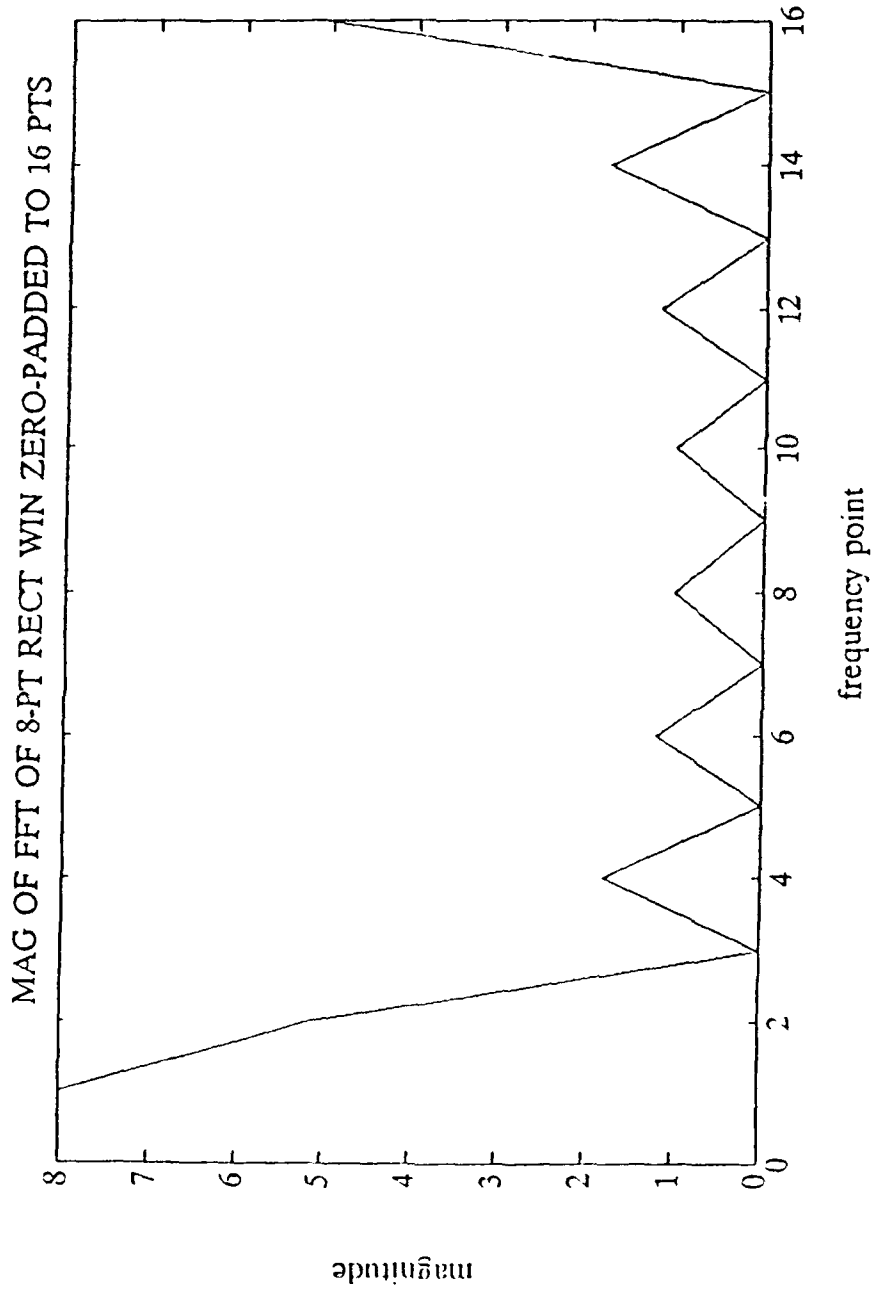


Figure 2. Magnitude of FFT of 8-point Rectangular Window Zero-padded to 16 Points

In this example, the bin centers of the FFT occur at integer multiples of f_s/N, which in this case is 64/64 or 1 Hz. Figure 3 shows that the spectral peak is well defined since the sinusoidal frequency lies exactly at a bin center and no zero-padding was performed prior to transformation. We do not see the side lobe structure of the digital sinc (transform of the rectangle function). Observe in Figure 4 what occurs when the frequency detected does not coincide with a bin enter. In this case, the frequency is 10.7 Hz, which is clearly not a bin center. The side lobes of the digital sinc function are now visible since we are not interrogating the sinc at points of its zero crossing. In addition, spectral leakage has smeared the signal power into the adjacent frequency bins. The end result is a much broader and less-pronounced main lobe (25 vs. 40 dB).

To illustrate the effects of zero-padding, let us now consider the situation in which the original 64-point data record has been zero-padded to 128. Now, regardless of whether or not the sinusoidal frequency is at a bin center, the side lobes of the digital sinc will now be visible as a result of the zero-padding (see Figures 5 and 6). The net effect will be a less pronounced main lobe due to the side lobes. In the case of f = 10.7 (Figure 6), the main lobe is flattened due to a combination of the sinc side lobes and spectral leakage.

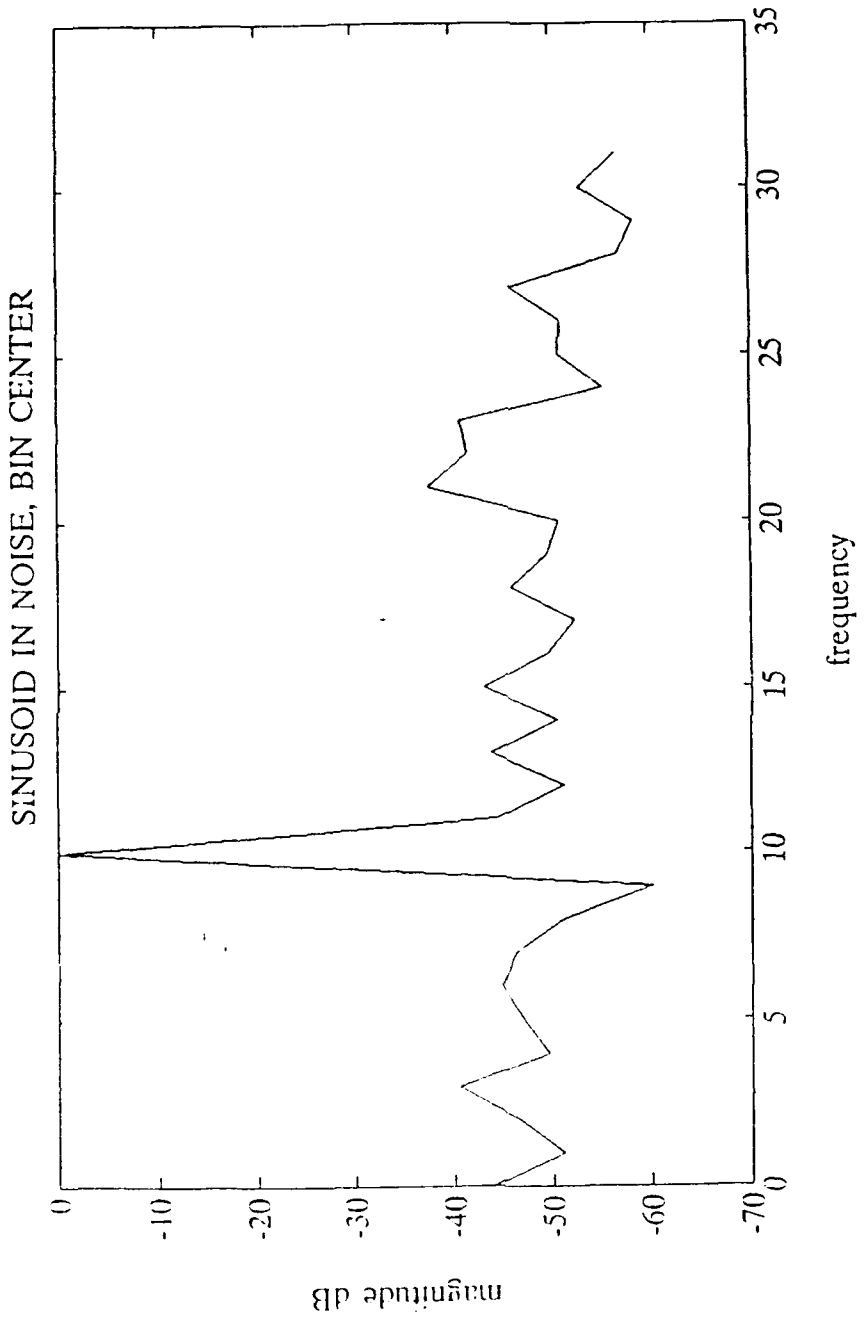


Figure 3. Periodogram of Sinusoid in Gaussian White Noise; f = 10.0 Hz (bin center); 64 points

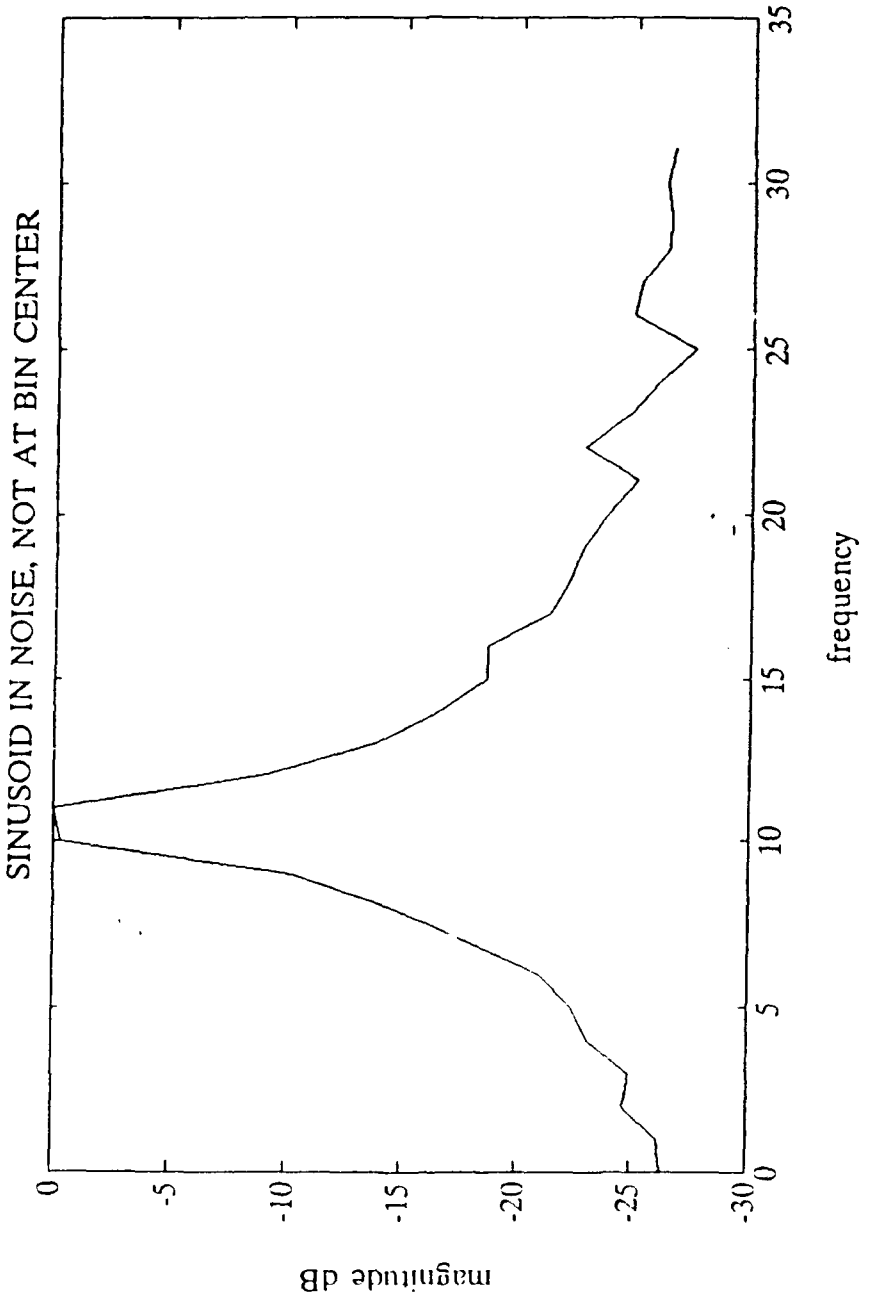


Figure 4. Periodogram of Sinusoid in Gaussian White Noise; f = 10.7 Hz (not a bin center); 64 points

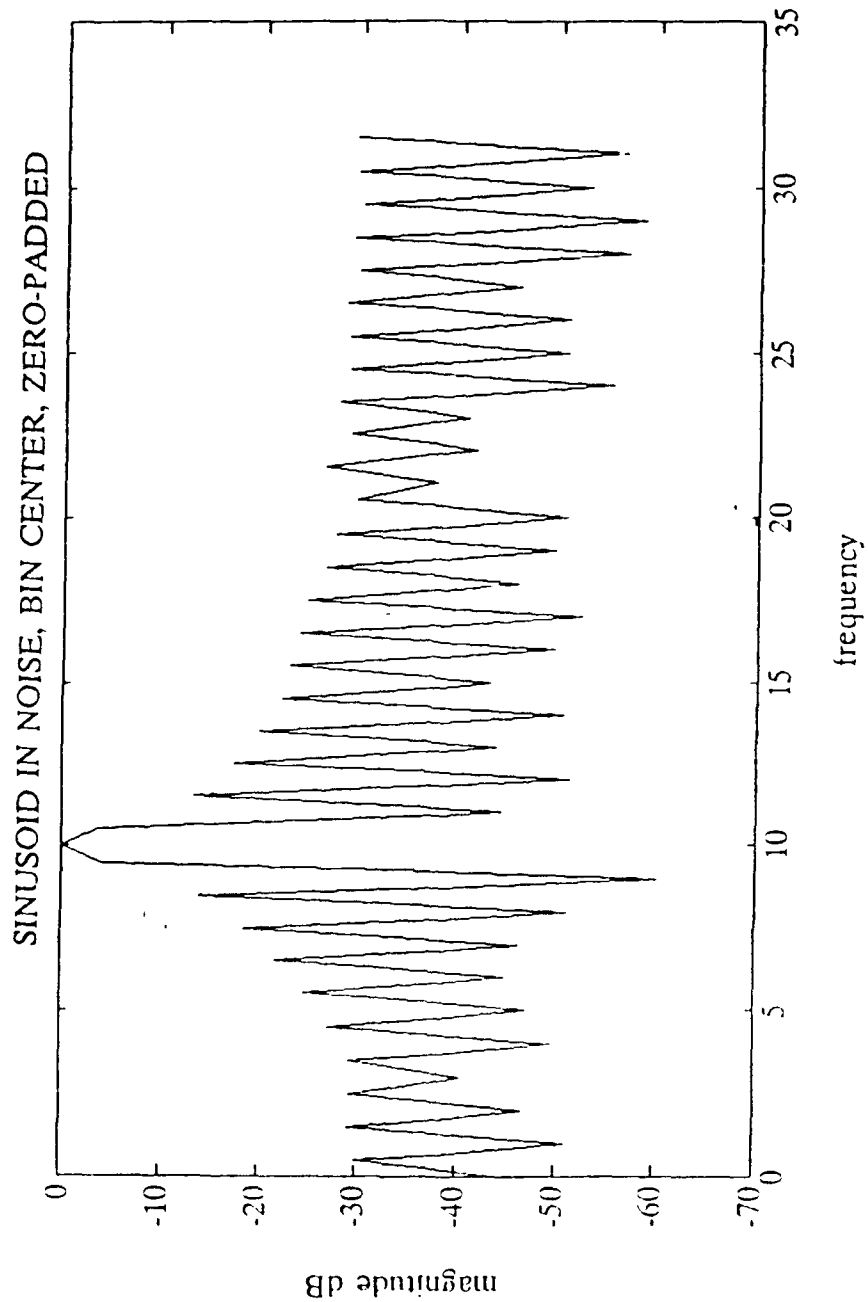


Figure 5. Periodogram of Sinusoid in Gaussian White Noise; f = 10.0 Hz (bin center); 64 points Zero-padded to 128

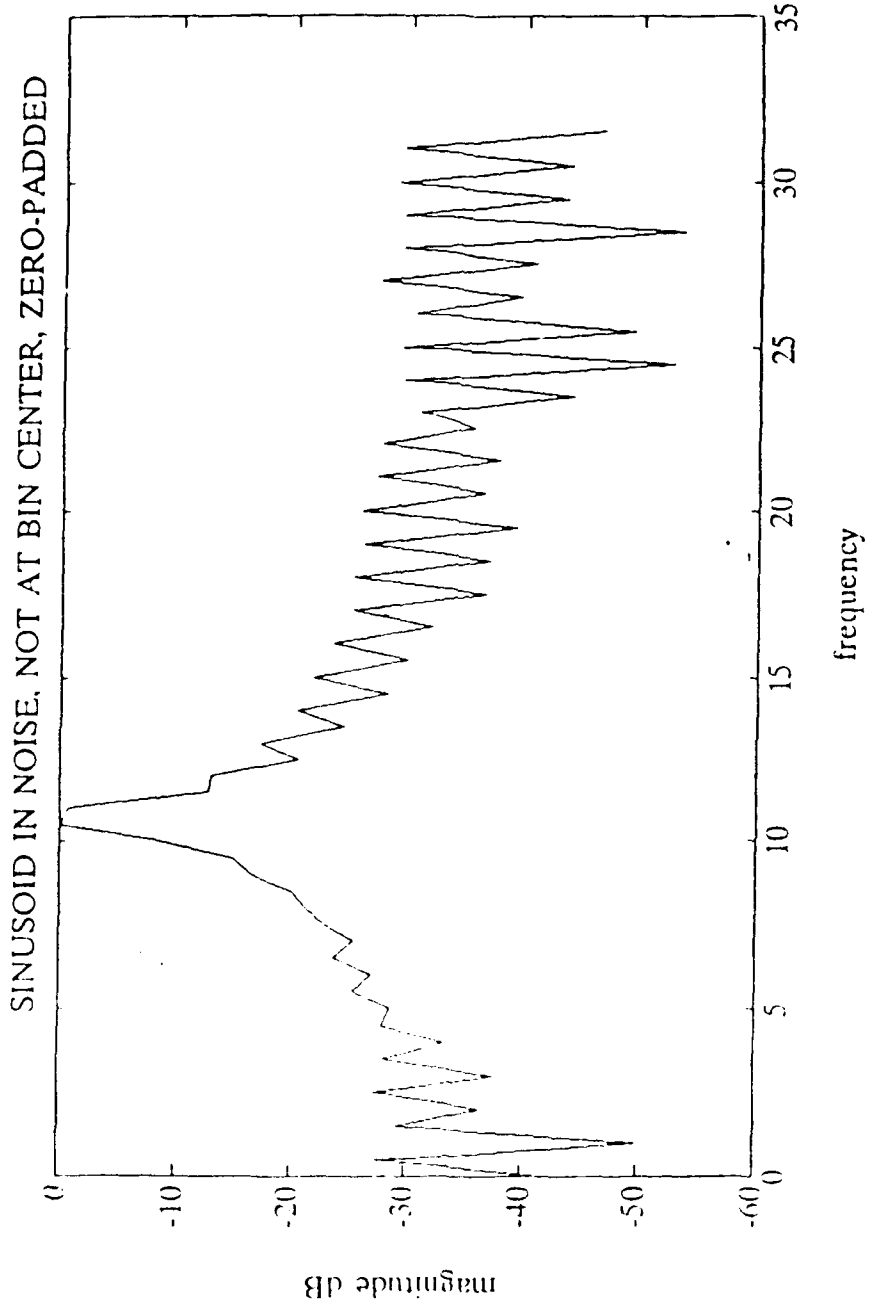


Figure 6. Periodogram of Sinusoid in Gaussian White Noise; f = 10.7 Hz (not a bin center); 64 points Zero-padded to 128

D. WINDOWS WITH NON-UNIFORM WEIGHTING

For comparison, the original 64-point data records for sinusoidal frequencies 10.0 and 10.7 Hz are weighted with a Hamming window prior to zero-padding and Fourier transformation (Figures 7 and 8). The Hamming window function, popular due to its good performance and ease of implementation, has a maximum side lobe level of -43 dB versus -13 dB for a rectangular window. The price paid for this side lobe suppression is increased main lobe width. The 3-dB main lobe width becomes 1.30 bins versus 0.89 bins for the rectangular window. The Hamming window is only one of many such functions. An exhaustive comparison of window functions and their use in spectral analysis is given by Harris [Ref. 1]. Many other windows, with even more dramatic reduction of side love levels, are possible. In all cases, however, the side effect is always a broadening of the main lobe with its associated reduction in spectral resolution [Refs. 1 and 3].

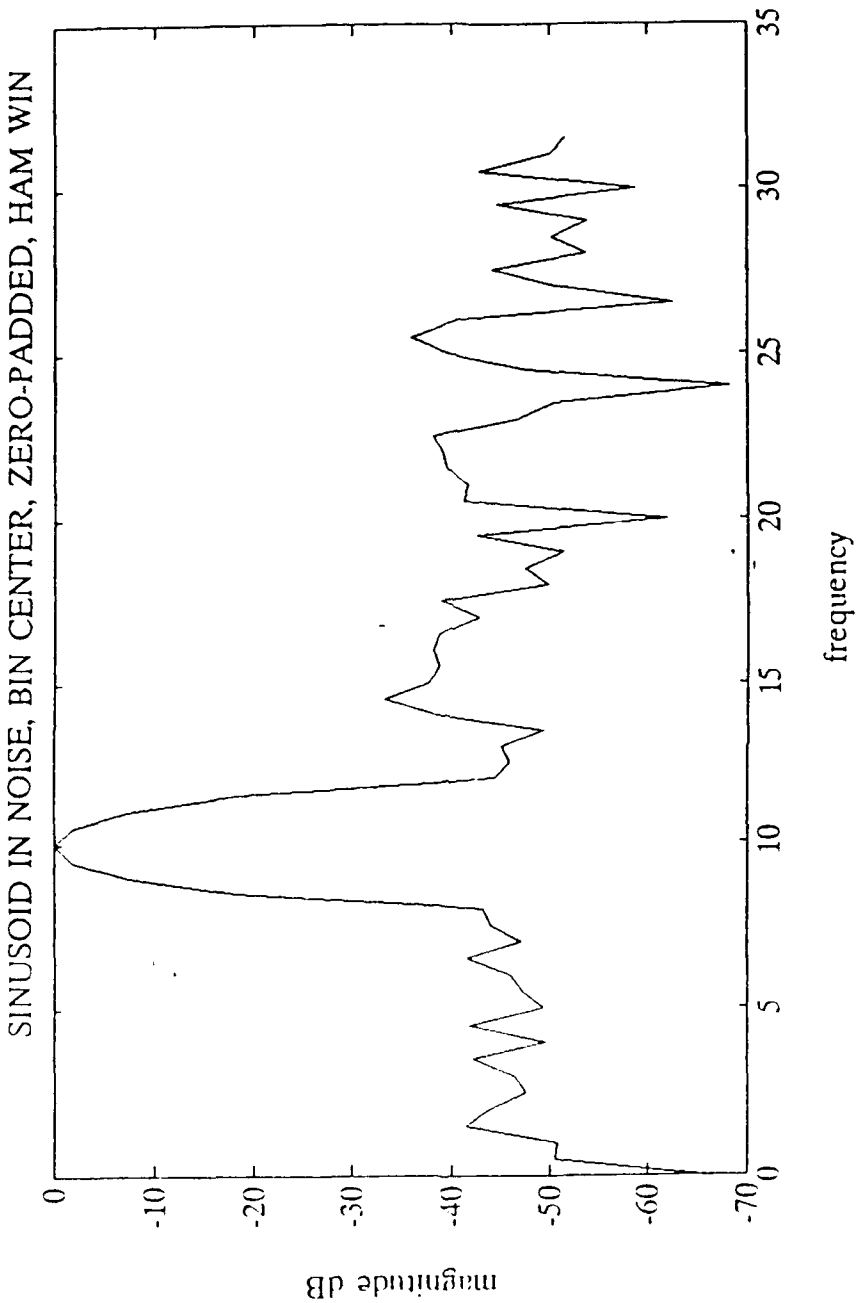


Figure 7. Periodogram of Sinusoid in White Noise; f = 10.0 Hz (not at bin center); 64 points Zero-padded to 128; Hamming window

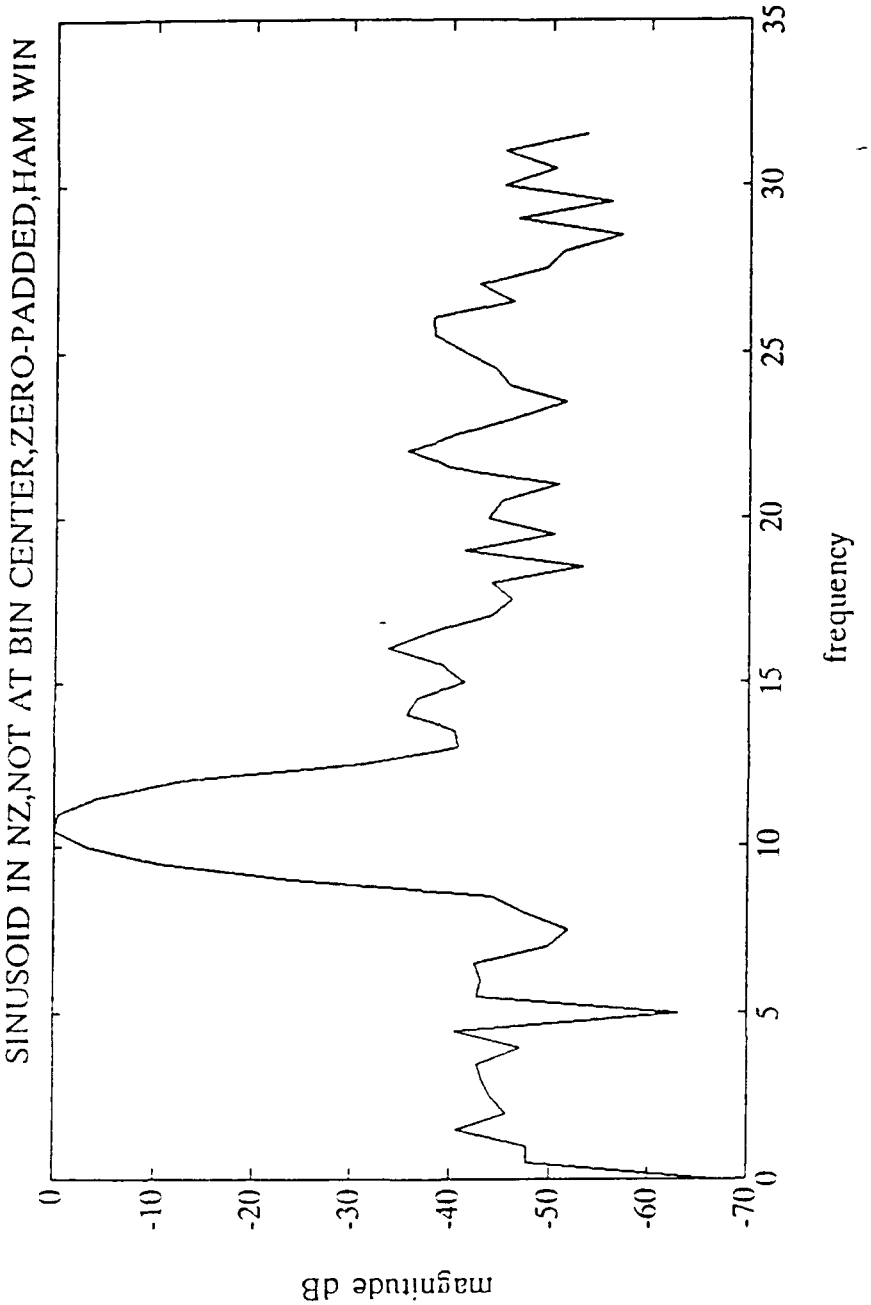


Figure 8. Periodogram of Sinusoid in White Gaussian Noise; f = 10.7 Hz (not at bin center); 64 points Zero-padded to 128; Hamming window

E. STATISTICAL PROPERTIES OF THE PERIODOGRAM

Consider a data record of samples of Gaussian white noise having zero mean and variance σ_x^2. The periodogram of this data will have a distribution which is chi-squared with two degrees of freedom. The reason for this is that the sampled Gaussian random process, denoted as x(n), has the distribution:

$$x(n) \sim N\left(0, \sigma_x^2\right) \quad . \tag{2.10}$$

For simplicity let us assume that the Fourier Transform of x(n) is normalized by 1/SQRT(N), where N is the size of the transform. Since the real and imaginary parts of the Fourier Transform of x(n), denoted as A(f) and B(f) respectively, are orthogonal linear combinations of x(n), it follows that A(f) and B(f) are mutually uncorrelated Gaussian random variables each having the distribution N(0, σ_x^2). The periodogram of x(n), $P_x(f)$, is defined as the sum of the squared real and imaginary parts of the Fourier Transform of x(n):

$$P_x(f) = A^2(f) + B^2(f) \quad . \tag{2.11}$$

The sum of the squares of two independent zero-mean normal variables is a chi-squared distribution with two degrees of freedom. The mean and variance of this distribution is given by:

$$E\left[P_x\left(f_p\right)\right] = 2\sigma_x^2; \qquad \text{all } p \tag{2.12}$$

$$Var\left[P_x\left(f_p\right)\right] = \begin{cases} 4\sigma_x^4; & p \neq 0, \dfrac{N}{2} \\ 8\sigma_x^4; & p = 0, \dfrac{N}{2} \end{cases} \tag{2.13}$$

where f_N denotes the sampling frequency [Ref. 4].

Proof of equation 2.13 for frequencies $p \neq 0, N/2$ is given in the following fashion:

Consider:

$$\begin{aligned} & P_x = A^2 + B^2 \\ \text{where} \quad & A \sim N\left(0, \sigma_x^2\right) \\ & B \sim N\left(0, \sigma_x^2\right) \end{aligned} \tag{2.14}$$

$$Var[P_x] = E\left[P_x^2\right] - \left(E[P_x]\right)^2. \tag{2.15}$$

We know that:

$$\begin{aligned} E\left[P_x^2\right] &= E\left[\left(A^2 + B^2\right)^2\right] \\ &= E\left[A^4 + 2A^2B^2 + B^4\right] \\ &= 8\sigma_x^4 \end{aligned} \tag{2.16}$$

and that from (2.12):

$$E[P_x] = 2\sigma_x^2$$

Therefore,

$$\begin{aligned} Var[P_x] &= E[P_x^2] - (E[P_x])^2 \\ &= 8\sigma_x^4 - (2\sigma_x^2)^2 \\ &= 4\sigma_x^4 \end{aligned} \tag{2.17}$$

F. PERIODOGRAM AVERAGING

The statistical properties of the periodogram may be improved by averaging a set of periodograms together. Assume that K independent data records are available, all for the interval $0 \leq n \leq (L - 1)$ and all are realizations of the same random process. The data is: $\{x_0(n), 0 \leq n \leq L - 1; x_1(n), 0 \leq n \leq L - 1; \ldots x_{k-1}(n), 0 \leq n \leq L - 1\}$. The averaged periodogram estimator is given by:

$$\hat{P}_{AV}(f) = \frac{1}{K} \sum_{m=0}^{K-1} \hat{P}_{PER_m}(f) \tag{2.18}$$

where $\hat{P}_{PER_m}(f)$ is the periodogram of the mth data set:

$$\hat{P}_{PER_m}(f) = \frac{1}{L} \left| \sum_{n=0}^{L-1} x_m(n) \exp(-j2\pi fn) \right|^2 \tag{2.19}$$

The mean value of the averaged periodogram will be the same as that of the periodogram based upon any of the individual data sets since periodograms for each set are independent and identically distributed. The variance of the periodogram will

be reduced by a factor of K as a result of the averaging operation. [Ref. 2]

$$Var\left[\hat{P}_{AV}(f)\right] = \frac{1}{K} Var\left[\hat{P}_{PER_m}(f)\right] \tag{2.20}$$

In actual practice, we seldom have independent data sets. It is more common to have one long data record of length N. A common technique is to segment the data into K non-overlapping blocks of length L, where N = KL. Since the blocks are contiguous, they cannot be uncorrelated for any process except white noise. Therefore, the actual variance reduction is bounded by a factor less than or equal to K. If the data are Gaussian white noise samples, the autocorrelation function of the data will decay rapidly and the blocks will be uncorrelated. Thus, the periodograms of the data segments will be independent and (2.20) will be accurate. [Ref. 3]. As an illustration, Figure 9 is the periodogram of 64 samples of Gaussian white noise (zero mean, variance 1/2000). Contrast this with Figure 10, which is the average of the periodograms of 5 independent 64-point data records obtained by segmenting a 320-point record of white noise samples with the same statistical properties. From (2.11), the predicted variance of the Figure 9 periodogram is $4(1/2000)^2 = 2.5 \times 10^{-7}$ for $p \neq 0, N/2$. From (2.18) we would expect a variance reduction by a factor of 1/N = 1/5 or 6.9 dB for the average periodogram.

The actual measured variance reduction between the single and averaged periodograms is 6.7 dB. A variation of this averaging scheme was proposed by Welch [Ref. 5] involving the application of a non-rectangular window function to each data segment and overlapping the segments (typically in a 4:1 ratio).

In interpreting spectral estimates, it is important to be able to discriminate between spectral detail due to statistical fluctuation and actual frequency content. A standard way of evaluating the goodness of a spectral estimator is via confidence intervals. A means of deriving a confidence interval for the averaged periodogram is described in References 3 and 6.

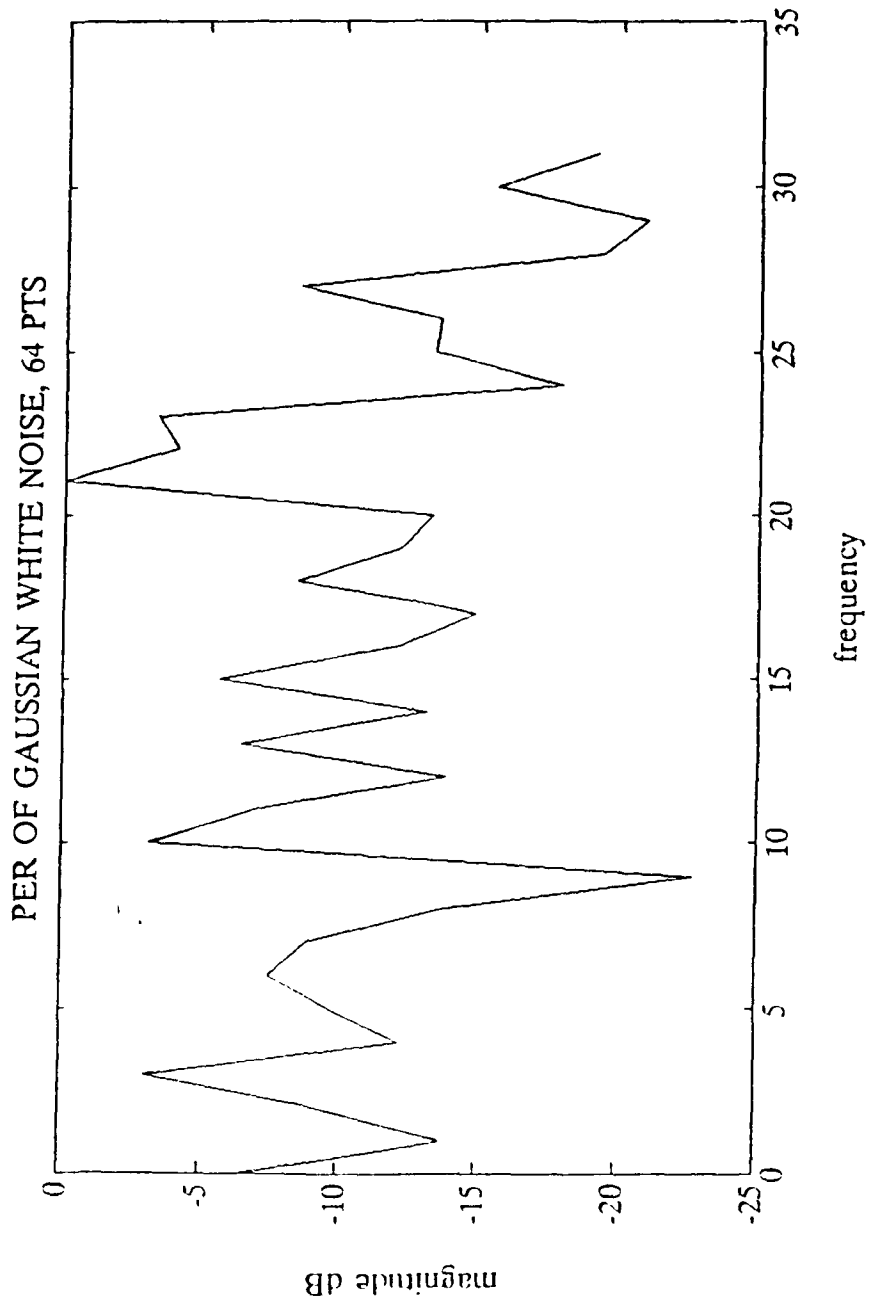


Figure 9. Periodogram of Gaussian White Noise 64 points, variance 1/2000

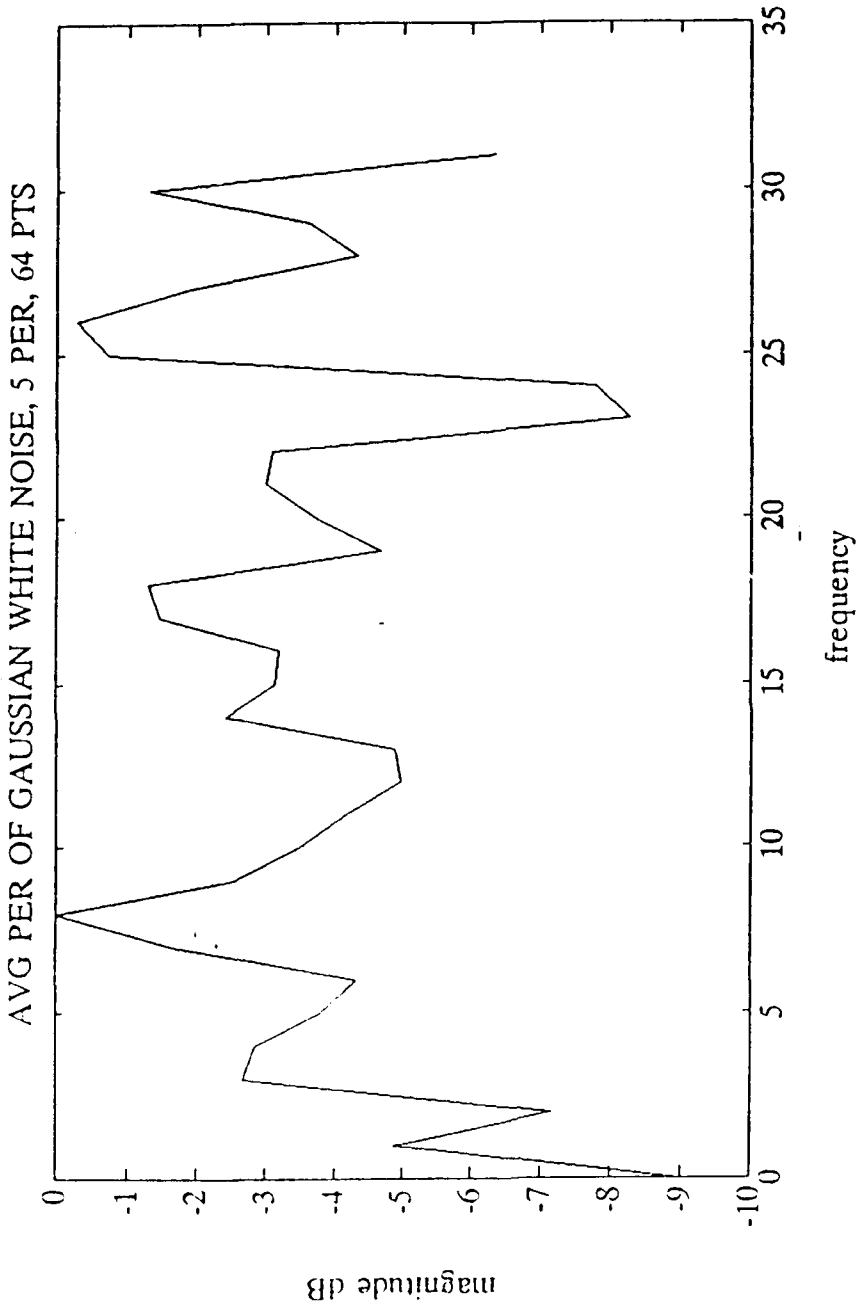


Figure 10. Average of 5 64-point Periodograms of Gaussian White Noise, Variance 1/2000

G. SPECTRAL SMOOTHING THE DANIELL PERIODOGRAM

Daniell suggested that a means of smoothing the fluctuations of the periodogram was to average over adjacent spectral frequencies. [Ref. 7] He proposed a modified periodogram estimate, $\hat{P}_D(f)$, in which each frequency spectral estimate was obtained by averaging over p spectral points on both sides of the frequency f under consideration. The Daniell Periodogram is given by:

$$\hat{P}_D(f_i) = \frac{1}{2p+1} \sum_{n=i-p}^{i+p} P(f_n) \qquad (2.21)$$

A generalization of this concept is to pass the sample spectrum through a low-pass filter with frequency response H(f). The Daniell periodogram may then be expressed as the convolution of the sample spectrum with a low-pass filter H(f) [Ref. 7].

$$\hat{P}_D(f) = \hat{P}(f) * H(f) \qquad (2.22)$$

The larger the p used, the greater the smoothing effect will be. As with other methods, the price paid for smoothing is a loss of resolution. Figure 11 shows the effect of Daniell's operation (p=2) on a spectral estimate in which the frequency of the test signal, 10.0 Hz, is at a bin center. Figure 12 shows Daniell's method performed on a spectral estimate where the frequency of the test signal, 10.7 Hz, is not at a bin center.

In summary, the FFT-based spectral estimation technique (i.e., the periodogram) remains popular due to its computational efficiency and good performance. Frequency resolution (in Hz) is proportional to the reciprocal of the length of the data measured in seconds. The ability to resolve closely-spaced signal components is degraded by a combination of side lobe effects and main lobe broadening. Side lobe suppression is possible through the use of non-uniformly weighted (non-rectangular) window functions but only at the cost of main lobe broadening. Despite these limitations and the advent of modern spectral estimation techniques such as parametric modeling, the periodogram remains the most popular spectral estimator as a result of its relative simplicity, robustness, and ease of implementation in real time.

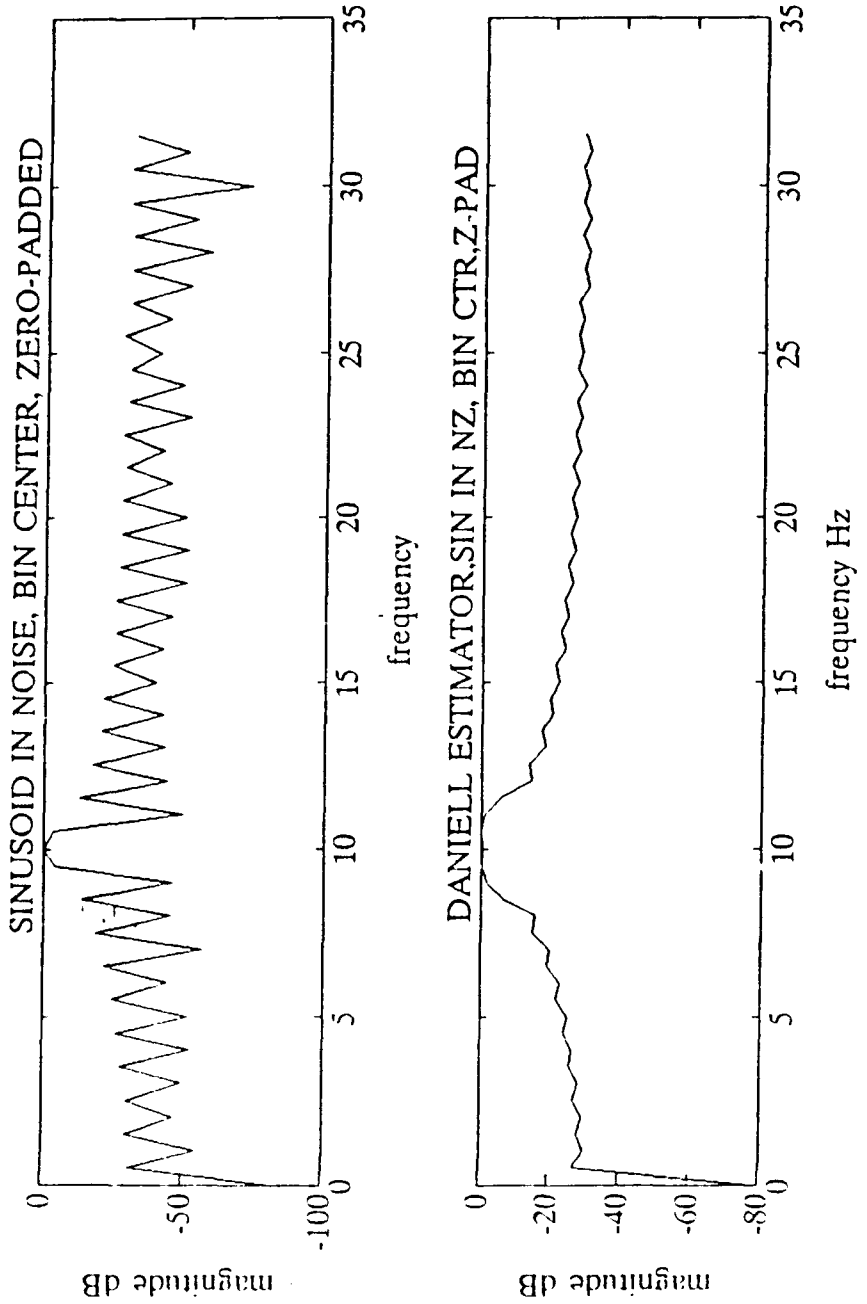


Figure 11. Daniell Spectral Estimator, p = 2
f = 10.0 Hz, 64 points Zero-padded to 128

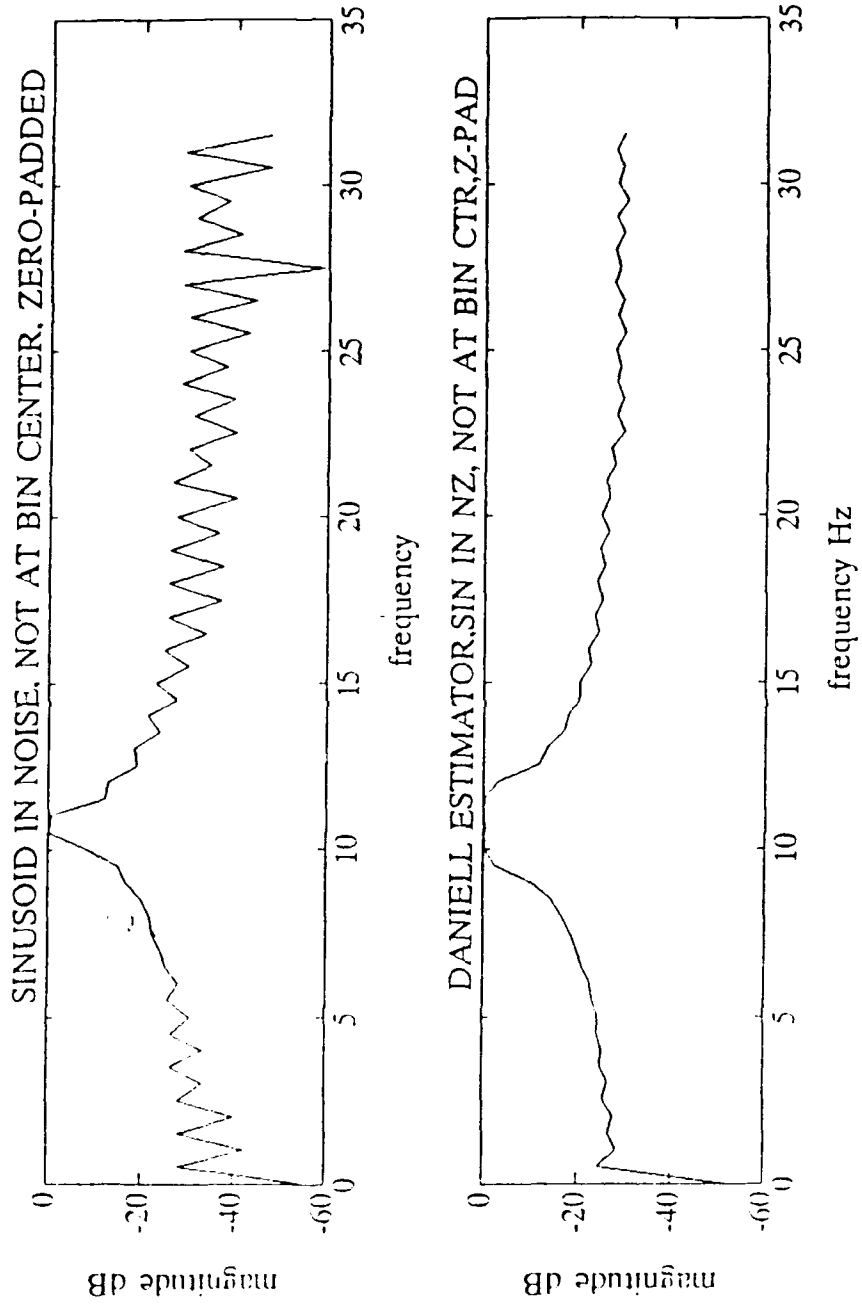


Figure 12. Daniell Estimator p = 2
f = 10.7 Hz, 64 points Zero-padded to 128

III. KALMAN FILTERING IN SPECTRAL ESTIMATION

A. BACKGROUND

A continuing problem with FFT-based spectral estimation schemes is the trade-off between spectral resolution and side lobe suppression. If a non-rectangular window function, i.e., the Hamming or Blackman window, is applied to time series data for the purpose of minimizing spectral side lobes, the side effect is a loss of resolution caused by the broadened mainlobe. In general, the better the side lobe suppression, the broader the main lobe. An extreme example is the minimum 4-sample Blackman-Harris window. The highest side lobe of this window which is 92 dB down from the main lobe peak. The cost of this level of side lobe attenuation is that the 3-dB bandwidth (main lobe) is 1.90 bins versus 0.89 bins for a rectangular window [Ref. 1]. What is proposed here is a novel application of the Kalman filter to the periodogram for the purpose of minimizing spectral sidelobe effects without the usual attendant loss of resolution.

The Kalman filter program demonstrated here was written by Dr. Roberto Cristi at the Naval Postgraduate School, Monterey, California in 1988. It is an implementation of the filtering algorithm first proposed by Kalman and Bucy [Refs. 8 and 9] and is now widely used in control system theory.

Cristi's program was originally developed to detect piecewise constant segments of time series data corrupted by noise.

A discrete time state-space system model is given by:

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Delta_B\underline{u}(k) + \Delta_F\underline{v}(k) \tag{3.1}$$

$$\underline{y}(k) = C\underline{x}(k) + \underline{w}(k) \tag{3.2}$$

where $\underline{x}(k)$ is the state vector, $\underline{u}(k)$ is the input, $\underline{v}(k)$ is an input disturbance, $\underline{y}(k)$ is the observed data and $\underline{w}(k)$ is the measurement noise. The discrete transition, input, input disturbance and observation matrices are Φ, Δ_B, Δ_F, and C respectively. The input disturbance and measurement noise are further specified by:

$$E\left[\underline{v}(k)\underline{v}^T(k+m)\right] = V_d\delta(m) \tag{3.3}$$

$$E\left[\underline{w}(k)\underline{w}^T(k+m)\right] = W_d\delta(m) \quad , \tag{3.4}$$

where V_d and W_d are covariance matrices.

The Kalman gain equations are given by:

$$\underline{P}(k+1|k) = \Phi\underline{P}(k|k)\Phi^T + \Delta_F V_d \Delta_F^T \tag{3.5}$$

$$\underline{K}(k+1) = \underline{P}(k+1|k)C^T\left[C\underline{P}(k+1|k)C^T + W_d\right]^{-1} \tag{3.6}$$

$$\underline{P}(k+1|k+1) = \left[I - \underline{K}(k+1)C\right]\underline{P}(k+1|k) \quad , \tag{3.7}$$

where $\underline{P}(k+1|k)$ denotes the covariance matrix at time k+1 given observations to time k and $\underline{K}$ is the Kalman gain matrix. The Kalman filter equations are given by:

$$\underline{\hat{x}}(k+1|k) = \Phi\underline{\hat{x}}(k) + \Delta_B\underline{u}(k) \qquad (3.8)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \underline{K}(k+1)\left[\underline{y}(k+1) - C\underline{\hat{x}}(k+1|k)\right], \qquad (3.9)$$

where $\underline{x}(k+1|k)$ denotes the estimate of x at time k+1 given observations to time k. Note that the initial condition $\underline{P}(0|0)$ must be specified in order to start the process:

$$\underline{P}(0|0) = E\left[(\underline{x}(0) - \underline{\hat{x}}(0))(\underline{x}(0) - \hat{x}(0))^T\right] . \qquad (3.10)$$

Equation 3.10 specifies the covariance matrix of the initial error. The covariance matrix is a measure of the confidence on the initial estimate $\underline{x}(0)$.

Consider the simple, one-dimensional problem of detecting a piecewise constant time series segment corrupted by noise, which was the original purpose of Cristi's program. The signal and its noisy observation are given by:

$$x(k+1) = x(k) \qquad (3.11)$$

$$y(k) = x(k) + w(k) , \qquad (3.12)$$

where w(k) is the corrupting noise.

Now define:

$$i(k) = y(k) - C\hat{x}(k) \tag{3.13}$$

where $\underline{x}(k)$ is the estimate of x, C is some constant, and i is the innovation sequence. The sequence i(k) represents new information not contained in the previous observations y(k-1), y(k-2),...y(0). Elements of the sequence i have the property:

$$L[i(k)y(k-m)] = 0 \quad \text{for all } m \geq 1 \tag{3.14}$$

Equation 3.14 states that each element of i is orthogonal to all past observations.

Using Baye's theorem, we can compute the probability of the observations (y(k), y(k-1),...y(0)) in the following fashion:

$$\Pr(y(k), y(k-1)\ldots y(0)) = \Pr(y(k)|y(k-1)\ldots y(0))\Pr(y(k-1)\ldots y(0)) \tag{3.15}$$

Utilizing the recursive property of this expression, we can write:

$$\Pr(y(k), y(k-1)\ldots y(0)) = \prod_{\tau=0}^{k} \Pr(y(\tau)|y(\tau-1)\ldots y(0))\Pr(y(0)); \quad k \geq 1 \tag{3.16}$$

Using the Orthogonality Principle, it can be shown [Ref. 10]:

$$\Pr(y(k)|y(k-1)\ldots y(0)) \sim N(C\hat{x}(k), CP(k)C^T + W_d) \quad , \tag{3.17}$$

where N denotes a normal distribution and W_d is the covariance matrix of the observation noise. At time k, it is then possible to compute the probability Pr(y(k)|y(k-1)...y(0)). If the data under examination consists of piecewise constant segments, then at each new observation two possibilities exist:

1) the current observation is a continuation of the last piecewise constant segment of data observed or

2) the current observation is the first element of a new segment of data with a constant value.

What is now required is a means of computing the probability that a transition between piecewise constant sections has occurred. Let us now define a parameter β as a means of quantifying the likelihood of a transition and a binary random variable γ as follows. If a transition has not occurred, then $\gamma = 0$ and the current observation is filtered using a Kalman filter updated with the current gain. If a transition has occurred, then $\gamma = 1$ and the current observation is filtered using reinitialized Kalman filter. Now define the probability density functions:

$$\Pr(\gamma(k)=0) = m\exp(\beta) \qquad (3.18a)$$

$$\Pr(\gamma(k)=1) = m\exp(-\beta) \ , \qquad (3.18b)$$

where $m = \frac{1}{\exp(\beta)+\exp(-\beta)}$ and k denotes the time index. Assume that each $\gamma(k)$ is an independent event.

$$\Pr(\gamma(0),\gamma(1)\ldots\gamma(N)) = \sum_{K=0}^{N}\Pr(\gamma(k)) \tag{3.19}$$

We desire to maximize the expression:

$$\Pr\left(\gamma(k)\middle|\underline{y}(k),\underline{\hat{\gamma}}(k-1)\right) \tag{3.20}$$

where $\underline{y}(k)$ is the vector of observations up to and including the current time k and $\underline{\hat{\gamma}}$ (k-1) is the vector of previous estimates of the binary random variable γ up to time (k-1). Equation 3.20 is the probability of a transition or non-transition (depending on γ = 0 or γ = 1), given present and previous observations $\underline{y}$ and estimates of γ. We desire to maximize Equation 3.20 with respect to $\underline{y}(k)$ and $\underline{\hat{\gamma}}(k-1)$ where

$$\begin{aligned}\underline{y}(k) &= \left[y(k),y(k-1)\ldots y(0)\right]\\ &= \left[y(k),\underline{y}(k-1)\right]\end{aligned} \tag{3.21}$$

and

$$\underline{\hat{\gamma}}(k-1) = \left[\hat{\gamma}(k-1),\hat{\gamma}(k-2)\ldots\hat{\gamma}(0)\right] \tag{3.22}$$

By Bayes' Theorem,

$$\begin{aligned}
&\Pr\left(\gamma(k)\middle|\underline{y}(k),\underline{\hat{\gamma}}(k-1)\right) \\
&\quad=\left[\frac{\Pr\left(\gamma(k),\underline{y}(k),\underline{\hat{\gamma}}(k-1)\right)}{\Pr\left(\underline{y}(k),\underline{\hat{\gamma}}(k-1)\right)}\right] \\
&\quad=\left[\frac{\Pr\left(y(k),\underline{y}(k-1),\gamma(k),\underline{\hat{\gamma}}(k-1)\right)}{\Pr\left(y(k),\underline{\hat{\gamma}}(k-1)\right)}\right] \\
&\quad=\frac{\left[\Pr\left(y(k)\middle|\underline{y}(k-1),\gamma(k),\underline{\hat{\gamma}}(k-1)\right)\cdot\Pr\left(\underline{y}(k-1),\gamma(k),\underline{\hat{\gamma}}(k-1)\right)\right]}{\Pr\left(y(k),\underline{\hat{\gamma}}(k-1)\right)} .
\end{aligned} \tag{3.23}$$

Assuming that $\gamma(k)$ is independent of $\underline{y}(k-1)$ and $\underline{\hat{\gamma}}(k-1)$, the second term in the numerator of (3.22) becomes:

$$\Pr\left(\underline{y}(k-1),\gamma(k),\underline{\hat{\gamma}}(k-1)\right)=\Pr\left(\gamma(k)\right)\Pr\left(\underline{y}(k-1),\underline{\hat{\gamma}}(k-1)\right) \tag{3.24}$$

Equation 3.23 is then maximized with respect to $\underline{y}(k)$ and $\underline{\hat{\gamma}}(k-1)$ by the expression:

$$\begin{aligned}
&\max\left\{\Pr\left(\gamma(k)\middle|\underline{y}(k),\underline{\hat{\gamma}}(k-1)\right)\right\} \\
&\quad=\max\left\{\Pr\left(y(k)\middle|\underline{y}(k-1),\gamma(k),\underline{\hat{\gamma}}(k-1)\right)\Pr\left(\gamma(k)\right)\right\} .
\end{aligned} \tag{3.25}$$

Define the likelihood function:

$$\begin{aligned}
&L\left(\gamma(k)\middle|\underline{y}(k),\underline{\hat{\gamma}}(k-1)\right) \\
&\quad=\ln\left\{\Pr\left(\gamma(k)\middle|\underline{y}(k),\underline{\hat{\gamma}}(k-1)\right)\right\} \\
&\quad=\ln\left\{\Pr\left(y(k)\middle|\underline{y}(k-1),\gamma(k),\underline{\hat{\gamma}}(k-1)\right)\right\}+\ln\left\{\Pr\left(\gamma(k)\right)\right\} .
\end{aligned} \tag{3.26}$$

Note that $\Pr(y(k) \mid \underline{y}(k-1), \gamma(k), \hat{\underline{\gamma}}(k-1))$ can be computed by the modified version of (3.17):

$$\begin{aligned} &\Pr\left(y(k)\middle|\underline{y}(k-1), \gamma(k), \hat{\underline{\gamma}}(k-1)\right) \\ &\quad = \Pr\left(y(k)\middle|y(k-1), y(k-2)\ldots y(k-l)\right) \\ &\quad \sim N\left(C\hat{x}(k), CP(k)C^T + W_d\right) \end{aligned} \tag{3.27}$$

where l is the time interval between the current sample k and the last detected transition. Equation 3.27 is evaluated for the two cases of an updated or reinitialized Kalman filter. The probability $\Pr(\gamma(k))$ can be computed via (3.18). Thus it is possible, given each observation and those proceeding it, to compute the probability that a transition between constant valued segments has or has not occurred.

By selection of the parameter β (see Equation 3.18), it is possible to adjust the likelihood that a transition will occur. The larger the β selected, the less likely the filter is to reinitialize. If "too small" a value of β is selected, the filter will reinitialize too often and little smoothing of the data will be done. If "too large" a β is used, the filter will become too insensitive to fluctuations in the data and will not reinitialize at all. In this case, transition points will not be detected and the original data will be obliterated (over-smoothed). Thus far, β must be determined heuristically depending upon the type of data under observation. In general, noisier data (more statistical

fluctuation) will require more smoothing and thus larger values for β.

Figures 13 - 16 demonstrate a test of the Kalman filter program on a square wave of amplitude ± 1 corrupted by Gaussian white noise of variance 0.40. Figure 13 shows the observed data with the uncorrupted signal. Figures 14 - 16 show the filtered data for β = 0.20, 4.00, and 50.00. Figure 14, β = 4.00, shows the case where a "good" value of β has been chosen. Note that the filter correctly detects the actual transitions in the observed data and reinitializes only at these points. As a result, accurate recovery of the original waveform is achieved. In contrast, Figure 15 shows what occurs when too small a β is selected. The filter becomes too sensitive to noise fluctuations, mistakenly interpreting many of them as transitions. The filter reinitializes too often (see lower plot of transition points) and less than optimum smoothing is performed. Figure 16 is the case where too large a β is used, rendering the filter too insensitive to transitions in the observations. After the initialization, the filter never detects a transition and thus never reinitializes. The result is the obliteration (over smoothing) of the true waveform.

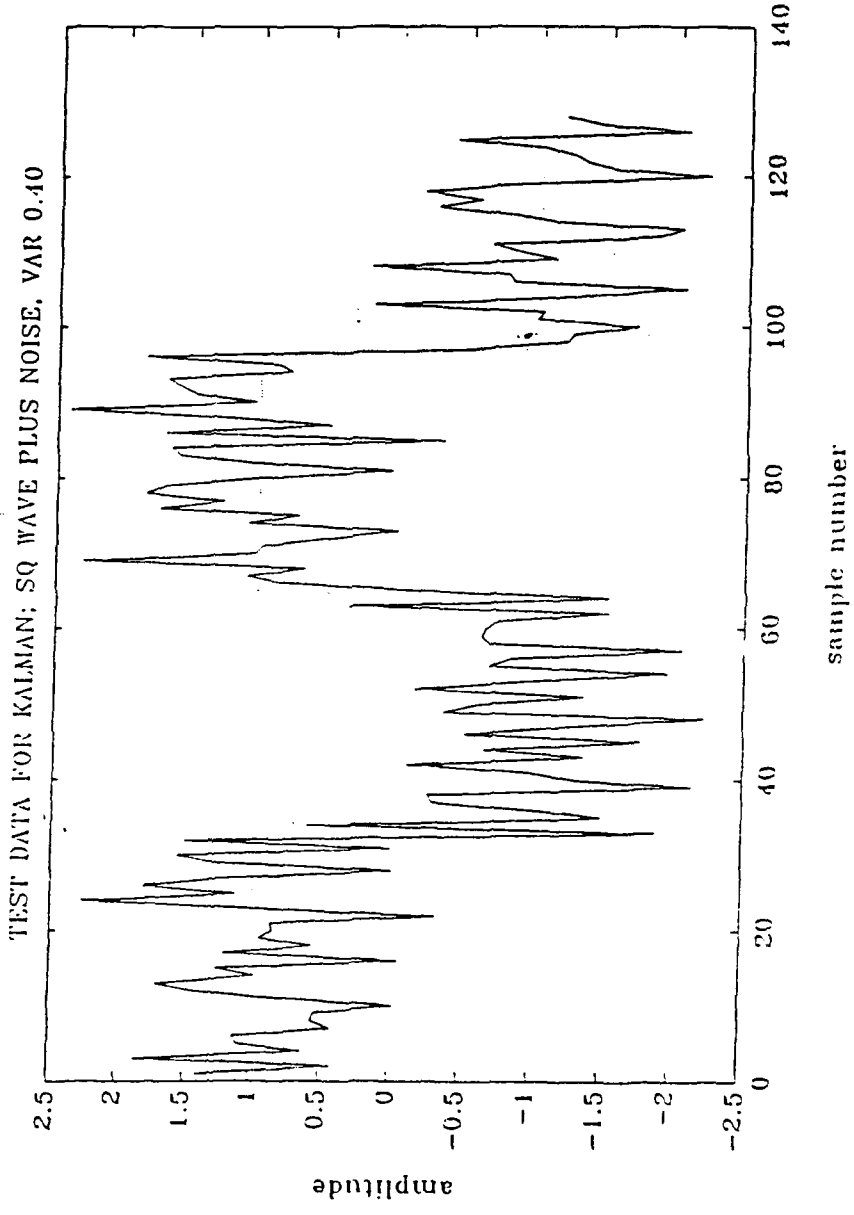


Figure 13. Test signal for Kalman Filter: Square Wave (±) Corrupted by Gaussian White Noise, Var = 0.40

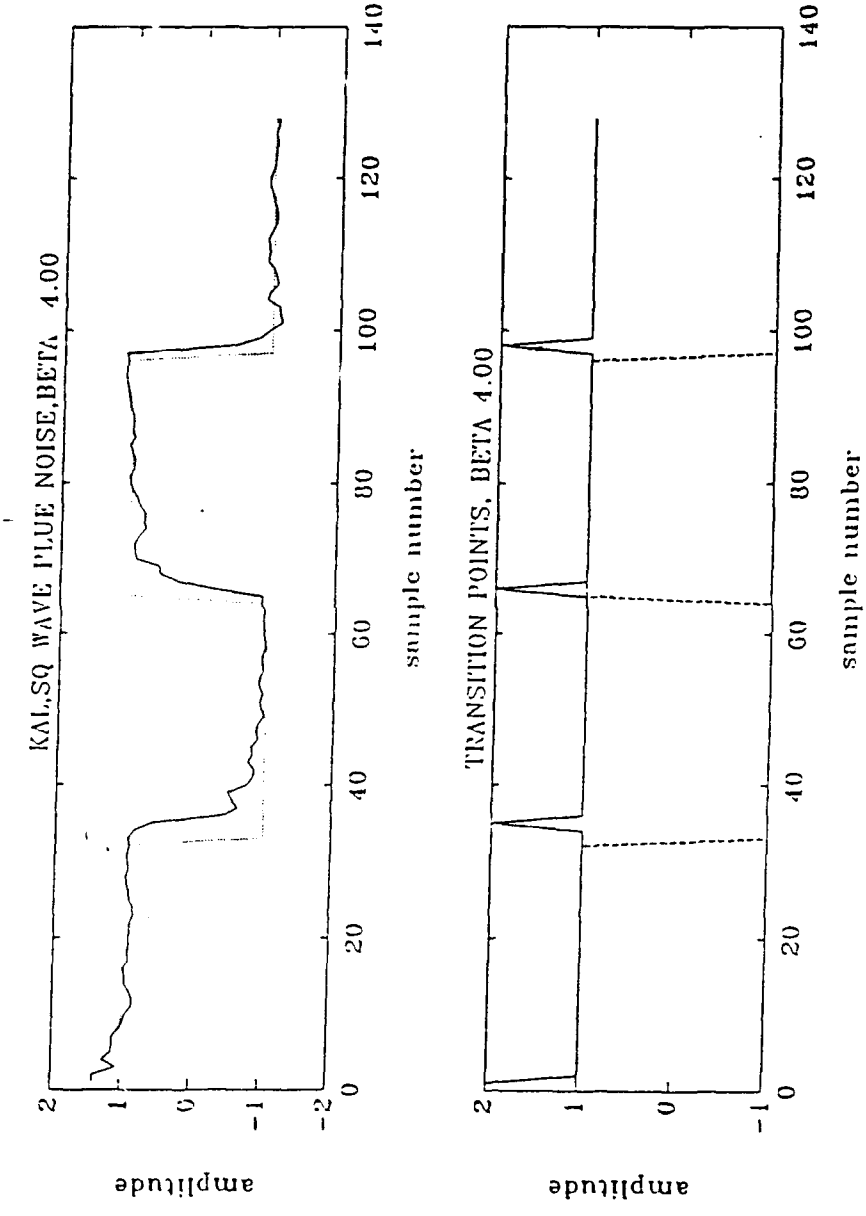


Figure 14. Output of Kalman Filter, Square Wave (±1) Plus Noise, $\beta = 4.00$

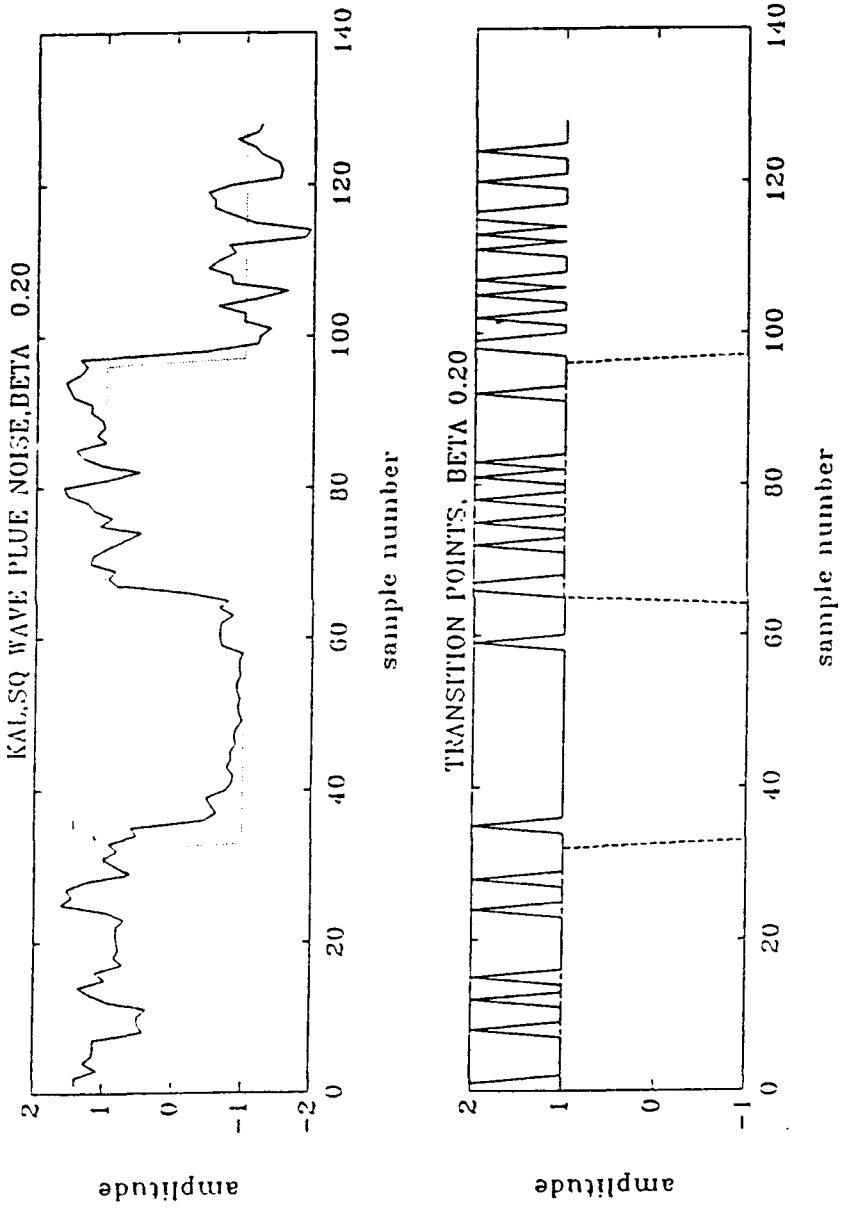


Figure 15. Output of Kalman Filter, Square Wave (±1) Plus Noise β = 0.20

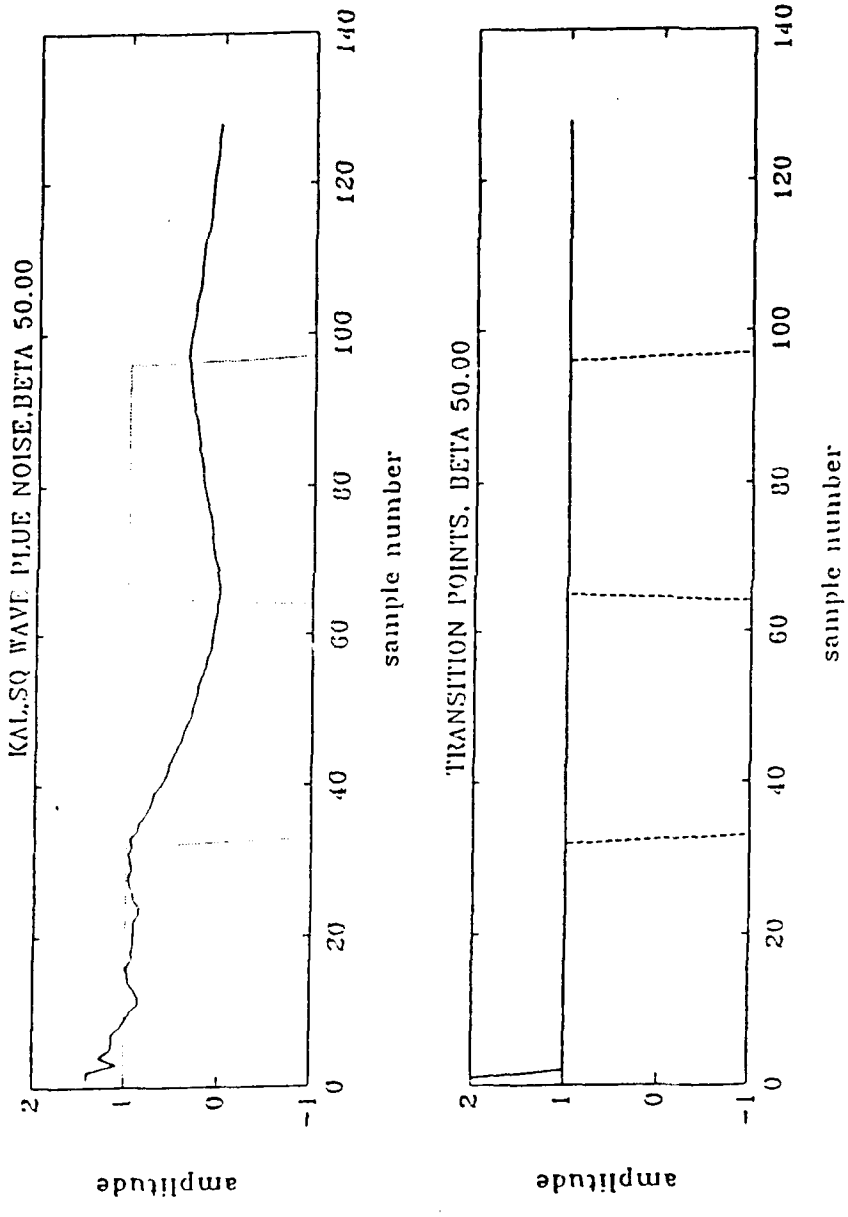


Figure 16. Output of Kalman Filter, Square Wave (±1) Plus Noise β = 50.00

B. KALMAN FILTERING APPLIED TO THE PERIODOGRAM

Now that the Kalman filter program has been demonstrated on a simple time series, the question arises: Can this algorithm be adapted for smoothing spectral data? The objective is to use the algorithm to smooth the periodogram spectral estimate with minimal broadening of the main lobe(s) of the dominant spectral responses. Ideally, an appropriate value for the parameter β is selected such that the noise portions of the periodogram are smoothed and transition points are detectable on either side of the spectral main lobe(s). The end result is a smoothed periodogram with the narrow main lobes of the original, unfiltered periodogram preserved. The noise "floor" out of which the signal peaks rise will be better defined and, hopefully, the frequency resolution of the original, unwindowed periodogram will be maintained.

The test signal used is a single sinusoid (unit amplitude) embedded in Gaussian white noise. The sinusoidal frequency is 10.7 Hz, which is not at a bin center. The signal is sampled at 64 Hz. A record of 128 data points is zero-padded to 256. The variance of the additive noise is varied to create input (time series) signal-to-noise ratios (SNRs) of -3, -6, -9, and -12 dB where SNR is as defined in Chapter II. Appendix C shows 10 different noise realizations at each SNR for a given value of β. The objectives of the investigation were three-fold:

1) To heuristically determine an "optimum" value for the parameter β, given the test conditions, at the different input SNR's.

2) To determine the input SNR of the time series (for 128 data points zero-padded to 256) at which the Kalman algorithm, given the "optimum" β, can reliably discriminate noise perturbation from signal peaks.

3) To determine if the Kalman algorithm preserves the spectral resolution of the unfiltered periodogram.

After many trials, it was determined that values for β in the range 100,000 to 700,000 provided the best compromise between undersmoothing and oversmoothing the spectral data. Within this optimum range, 100,000 causes the least smoothing and 700,000 the most. The the lowest input SNR (time series) at which reliable signal discrimination was achieved was -6 dB. At -6 dB, β = 300,000 gave generally good results. Signals could be detected at SNRs (time series) as low as -12 dB, depending on the noise realization (see Appendix C).

The consequences of too large or too small a β in the frequency domain are analogous to the time series example depicted in Figure 14 - 16. Figure 17 illustrates the results of the Kalman filter at an input SNR (time series) of -6 dB (128 data points zero-padded to 256), β= 300,000. Note that the single spectral peak due to the sinusoid has been left largely unaltered (unbroadened) and that we have successfully smoothed the noise portion of the periodogram. The filtered

periodogram, Figure 17, more closely approximates the ideal model of a spectral peak protruding up through a noise floor of constant value. In all cases, the Kalman filter was applied to periodograms of unwindowed (rectangular window) data. This resulted in the most narrow of possible main lobes and provides the highest resolution. For comparison, a Hamming window was applied to the time series data prior tc transformation (Figure 17). Some spectral smoothing is apparent along with the expected main lobe broadening. The noise floor is far less apparent than in the Kalman filtered periodogram. Figures 18 through 20 demonstrate the effects of varying β for a given noise realization, data/transform length and input SNR. In Figure 18, using β= 10.0, we obtain some smoothing, but the end result is little improvement over what is obtained with the Hamming window (Figure 18). Note that even at this low value of β, we have smoothed the spectra and preserved the narrowness of the main lobe. Figure 19, $\beta = 2.00 \times 10^{6}$, illustrates the effect of a β which is too large for the given input SNR and noise realization. Note the tapering effect on the higher frequency side of the main lobe. This is a symptom of over-filtering (over-smoothing) caused by too large a value of β. A smaller, closer-to-ideal β would have caused the filter to reinitialize after the peak and thus preserve the sharp down-transition of the original periodogram. In this case, the filter did not reinitialize and smoothed the higher frequency side of the main lobe.

Whenever this tapering effect is encountered, better results (sharper main lobe) can usually be obtained by reducing β. Figure 20, $\beta = 5.00 \times 10^6$, demonstrates obliteration of the original spectra caused by a β which is grossly too large. Figures B.1 and B.2 in Appendix B show the effect of varying β over a wide range for a given data record length, transform length, input SNR, and noise realization.

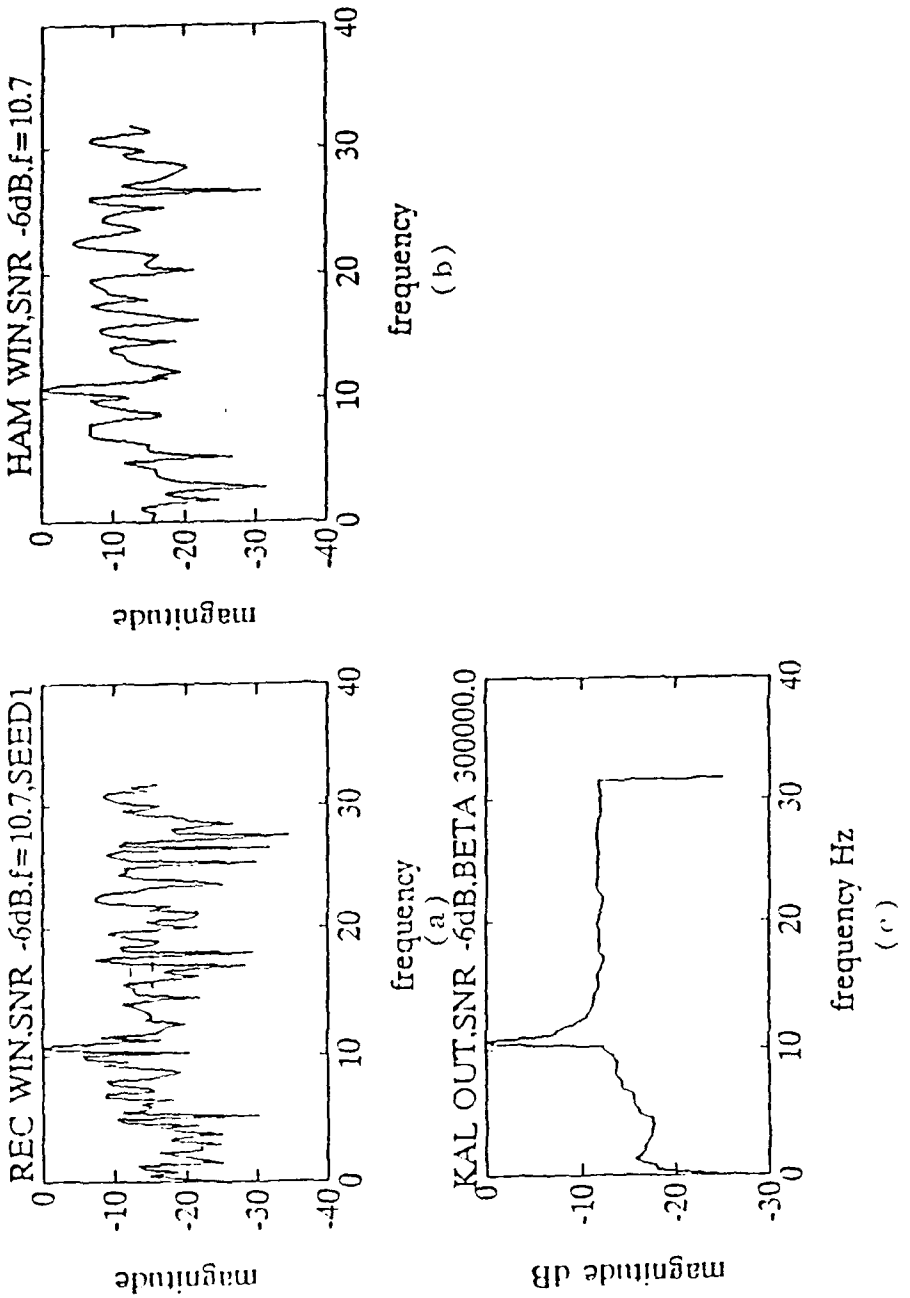


Figure 17. Sinusoid (f = 10.7 Hz) Plus Noise, SNR -6 dB
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 300000

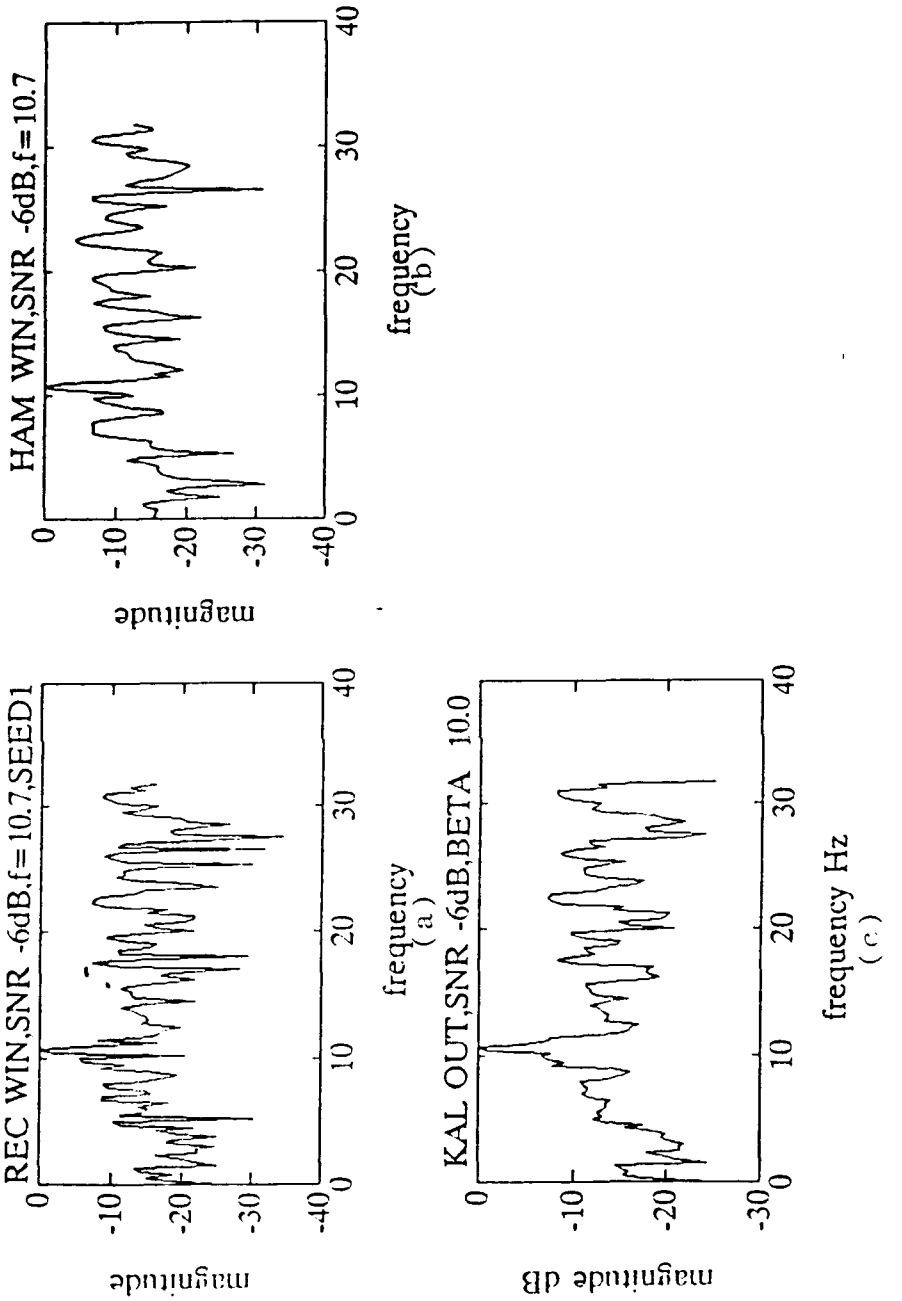


Figure 18. Sinusoid (f = 10.7 Hz) Plus Noise, SNR -6 dB
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 10.0

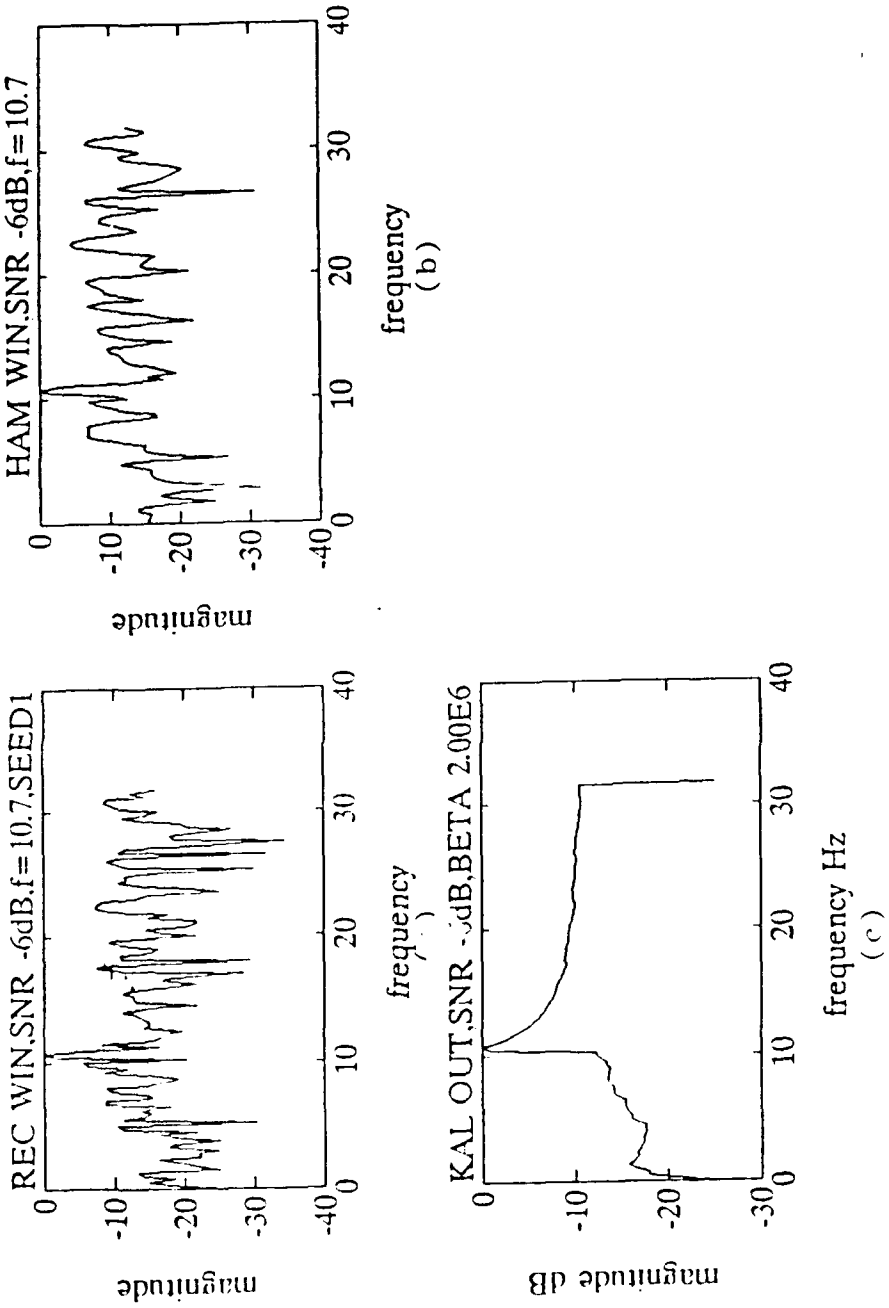


Figure 19. Sinusoid (f = 10.7 Hz) Plus Noise, SNR -6 dB
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 2.00 E6

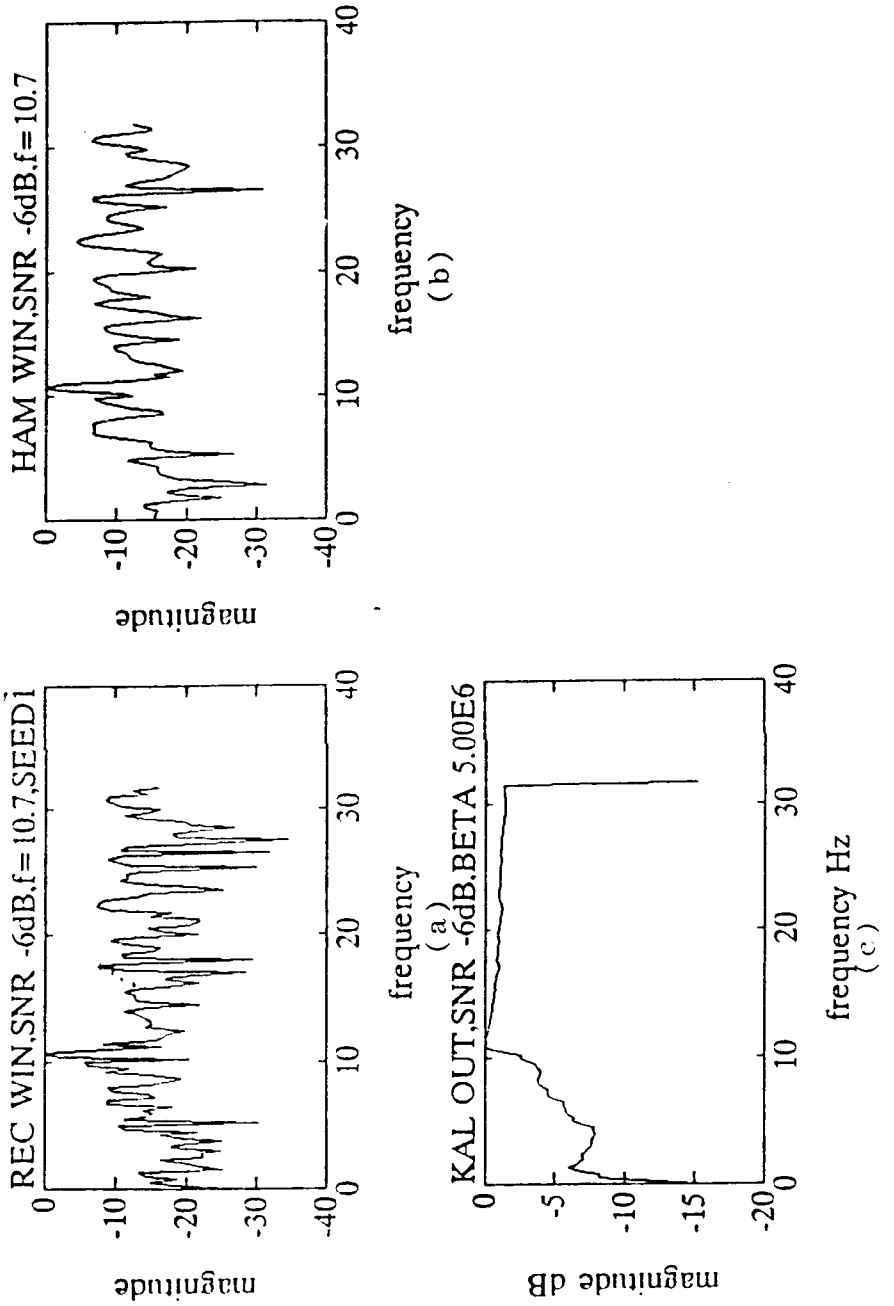


Figure 20. Sinusoid (f = 10.7 Hz) Plus Noise, SNR -6 dB
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 5.00 E6

C. EFFECTS ON SPECTRAL RESOLUTION

In order to evaluate the effects of the Kalman filter on spectral resolution, a second spectral component was added to the test data. For the test periodogram, bin width is f_s/N = 64/128 = 0.5 Hz. Note that we used N = 128, the data record size, and not N = 256, the transform length. As stated in Chapter II, zero-padding does not improve frequency resolution. It merely allows us to interpolate more frequency points. Initially, a second sinusoid (also unit amplitude) at 13.9 Hz was introduced. The frequency 13.9 Hz, like 10.7 Hz, is not a bin center and is many bin widths separate from 10.7 Hz. With β= 30,000, the Kalman filter successfully discriminated the signal peaks from the background noise (see Figure 21). Next, the second sinusoidal frequency was brought in to 11.2 Hz, one binwidth separation from the original signal at 10.7 Hz. This is close to the 0.89 binwidth resolution limit of the rectangular window. The two peaks are clearly visible in the unfiltered periodogram (see Figure 22). After filtering by the Kalman filter, the spectral estimate is smoothed and the resolution of the original periodogram is preserved as evidenced by the two still-visible spectral peaks (see Figure 22).

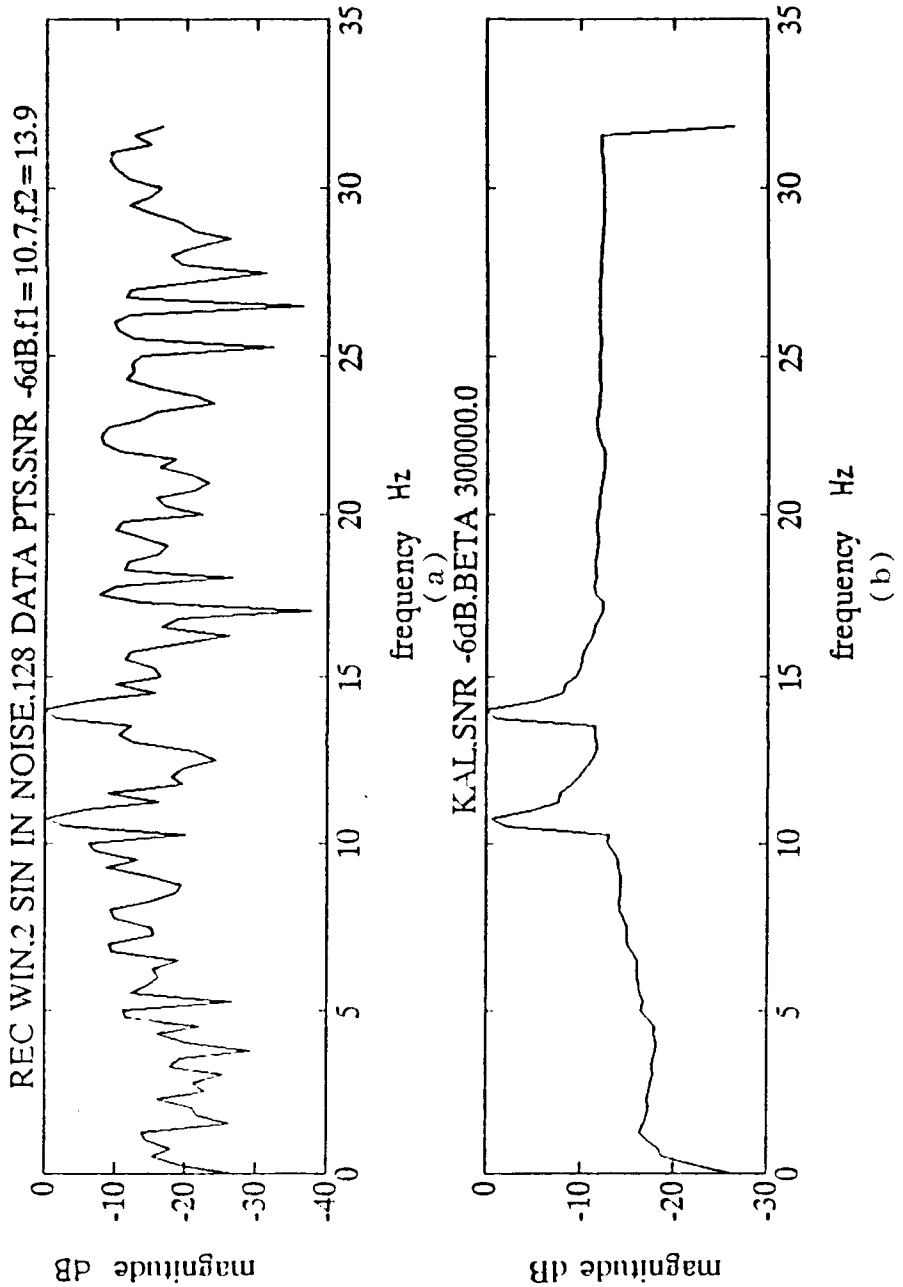


Figure 21. Two Sinusoids (f = 10.7, 13.9 Hz) Plus Noise, SNR -6 dB
(a) Periodogram (rectangular window)
(b) Kalman Filter Output β = 300000

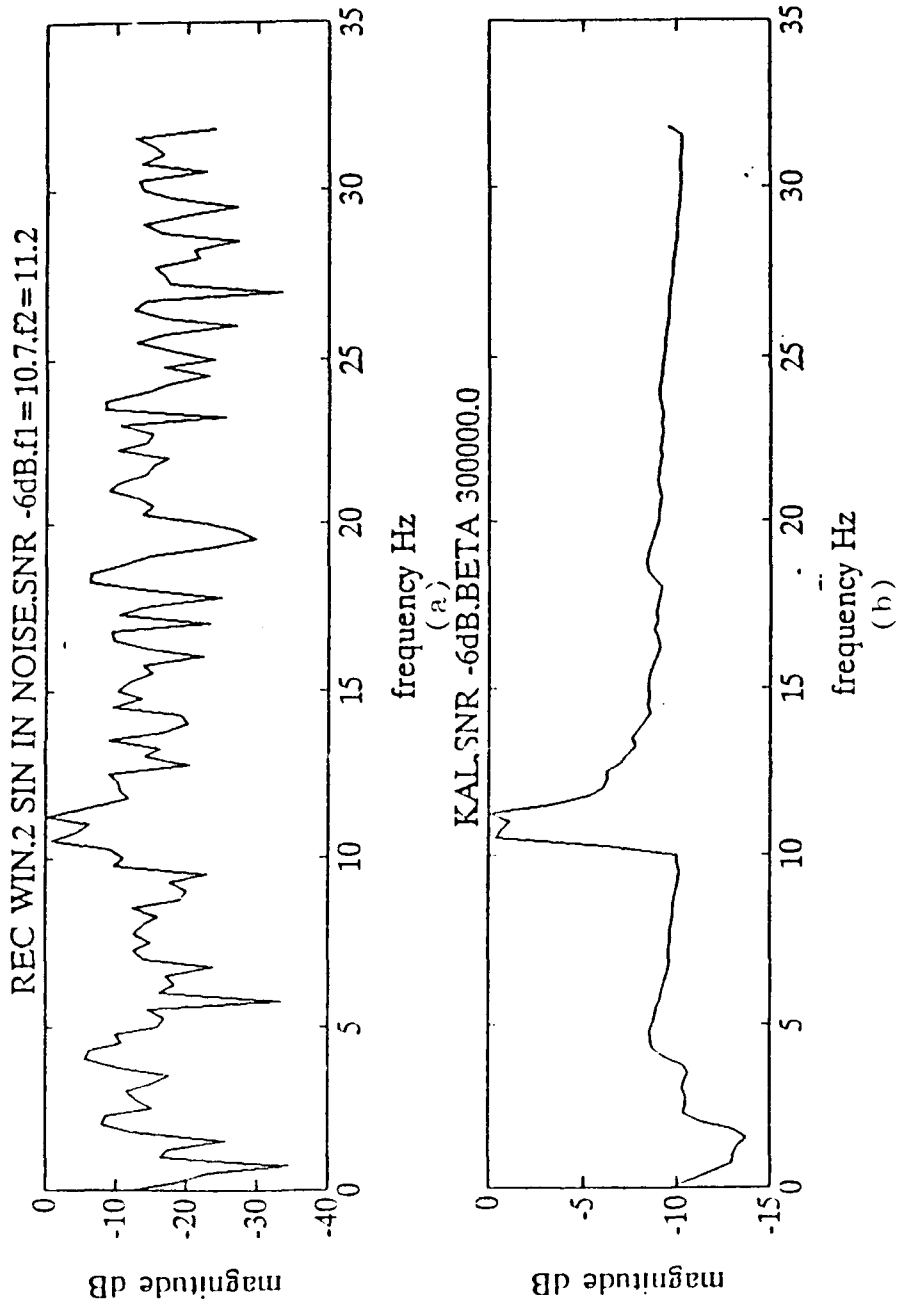


Figure 22. Two Sinusoids (f = 10.7, 11.2 Hz) Plus Noise, SNR -6 dB
(a) Periodogram (rectangular window)
(b) Kalman Filter Output β = 300000

D. THE NOISE-ONLY AND SIGNAL-ONLY CASES

The effects of the Kalman filter on noise-only and signal-only periodograms was tested. Figures 23-25 show the Kalman filter applied to three different realizations of Gaussian white noise, zero mean, 0.5 variance. As before, 128 sample points were zero-padded to 256. Using our "ideal" β of 300,000, no sharp spectral peaks were discriminated. This was to be expected since no dominant spectral component was present. Contrast these results with Figure 26, which is the Kalman filter applied to signal-only data. In Figure 26a, the characteristic sinc function, translated up to the sinusoidal frequency 10.7 Hz, is visible. Figure 26b shows the well-known smoothing and broadening effects of the Hamming window. In Figure 26c, with β = 300000, the Kalman filter smoothed the side lobe structure of the sinc and preserved the narrow spike of the main spectral peak.

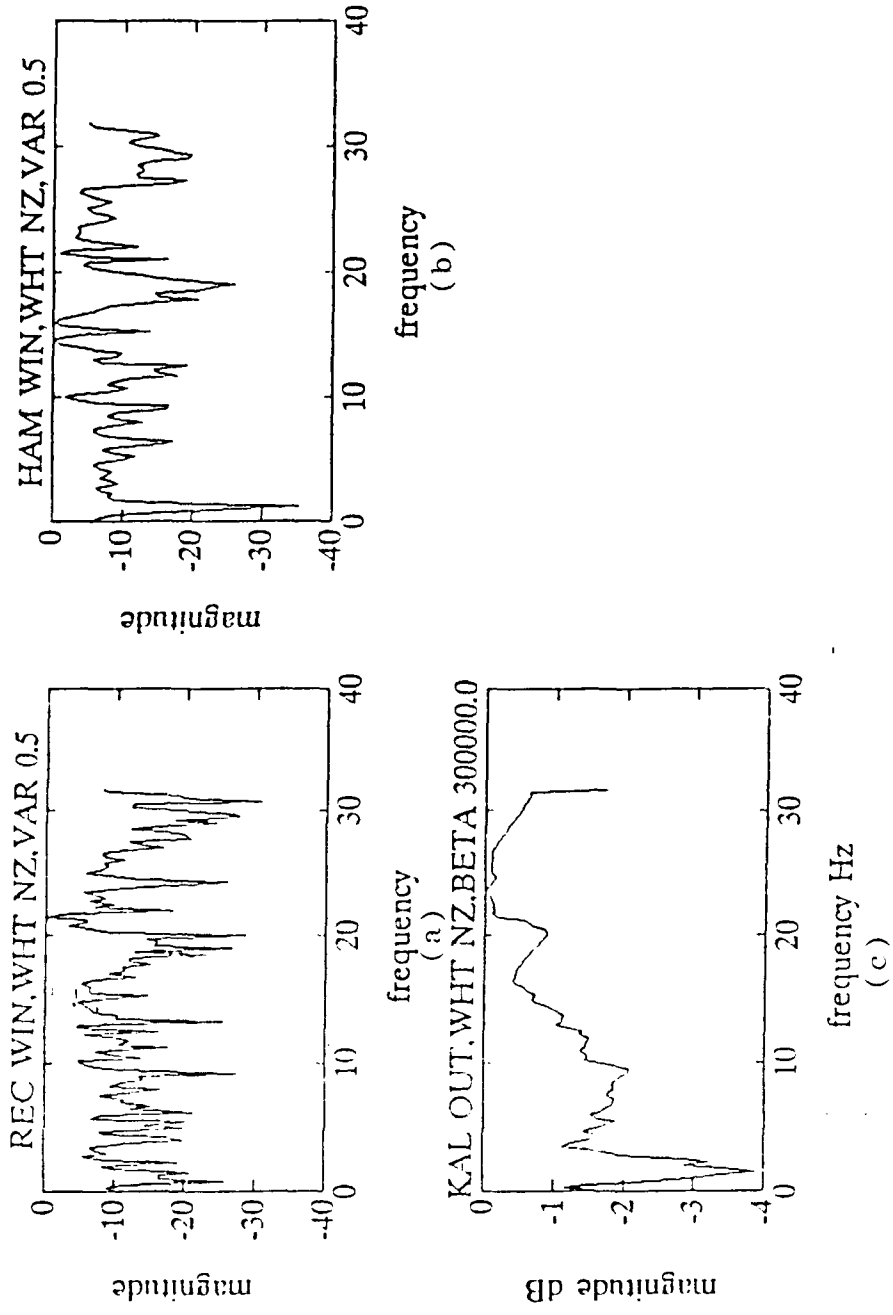


Figure 23. Noise-Only, Var 0.5, Realization 1
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 300000

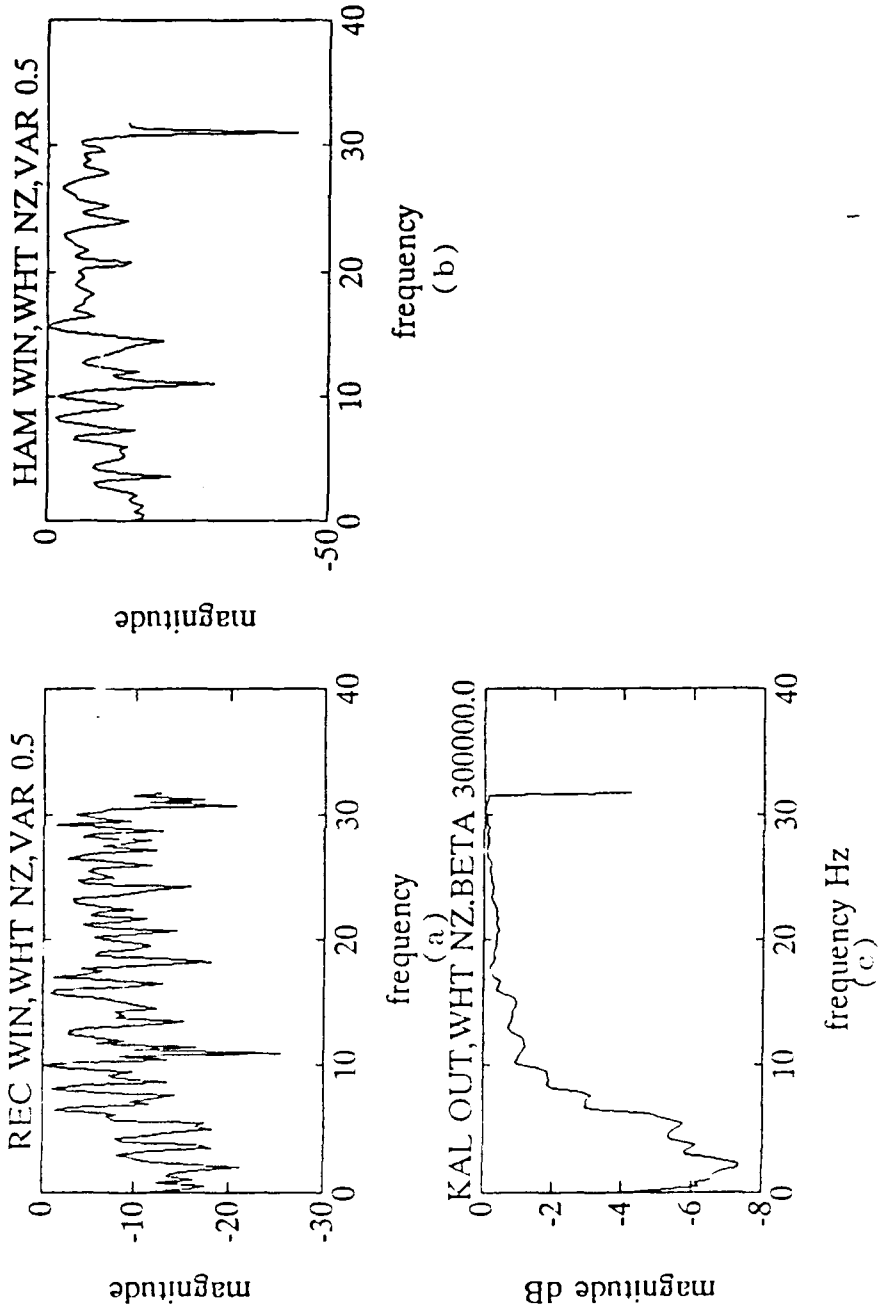


Figure 24. Noise-Only, Var 0.5, Realization 2
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 300000

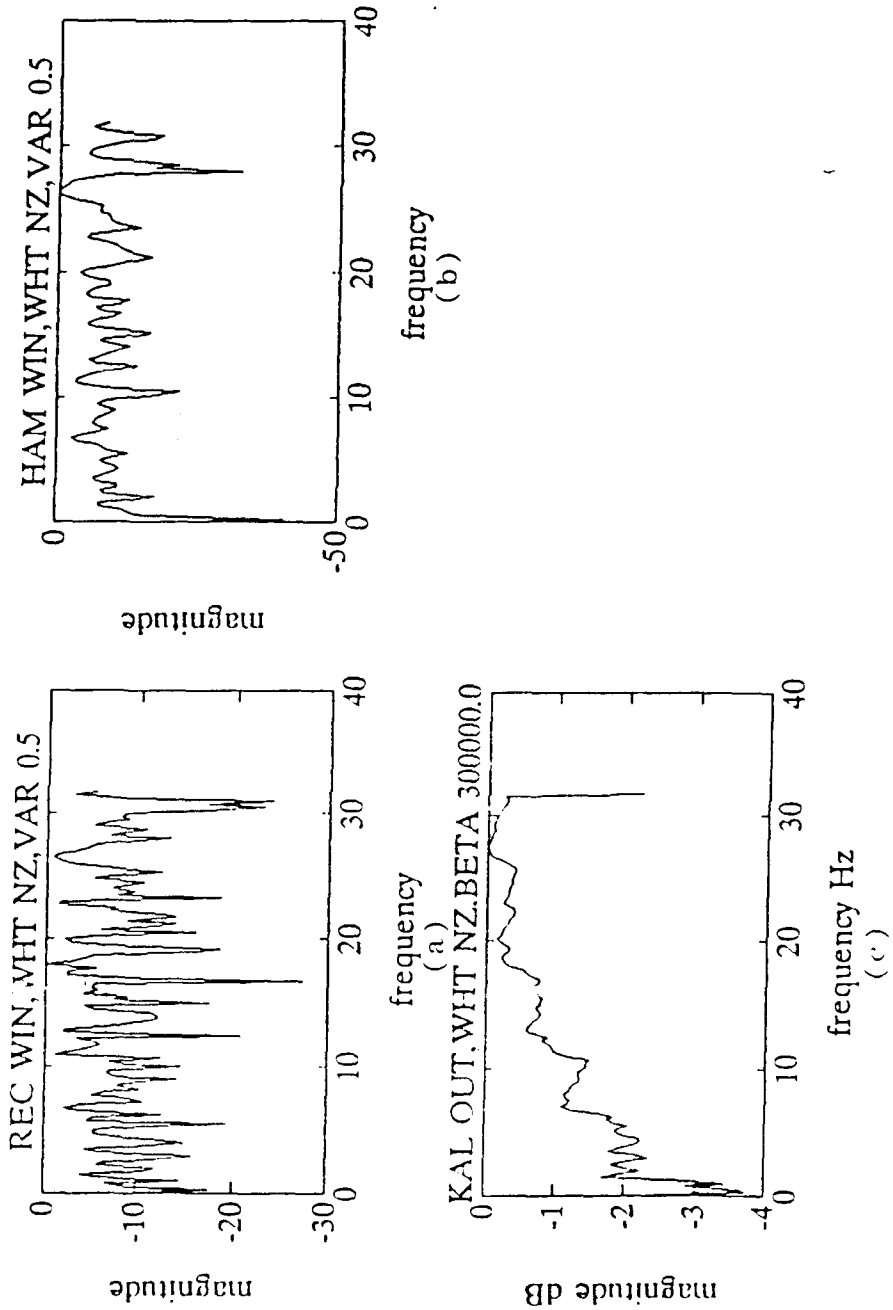


Figure 25. Noise-Only, Var 0.5, Realization 3
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 300000

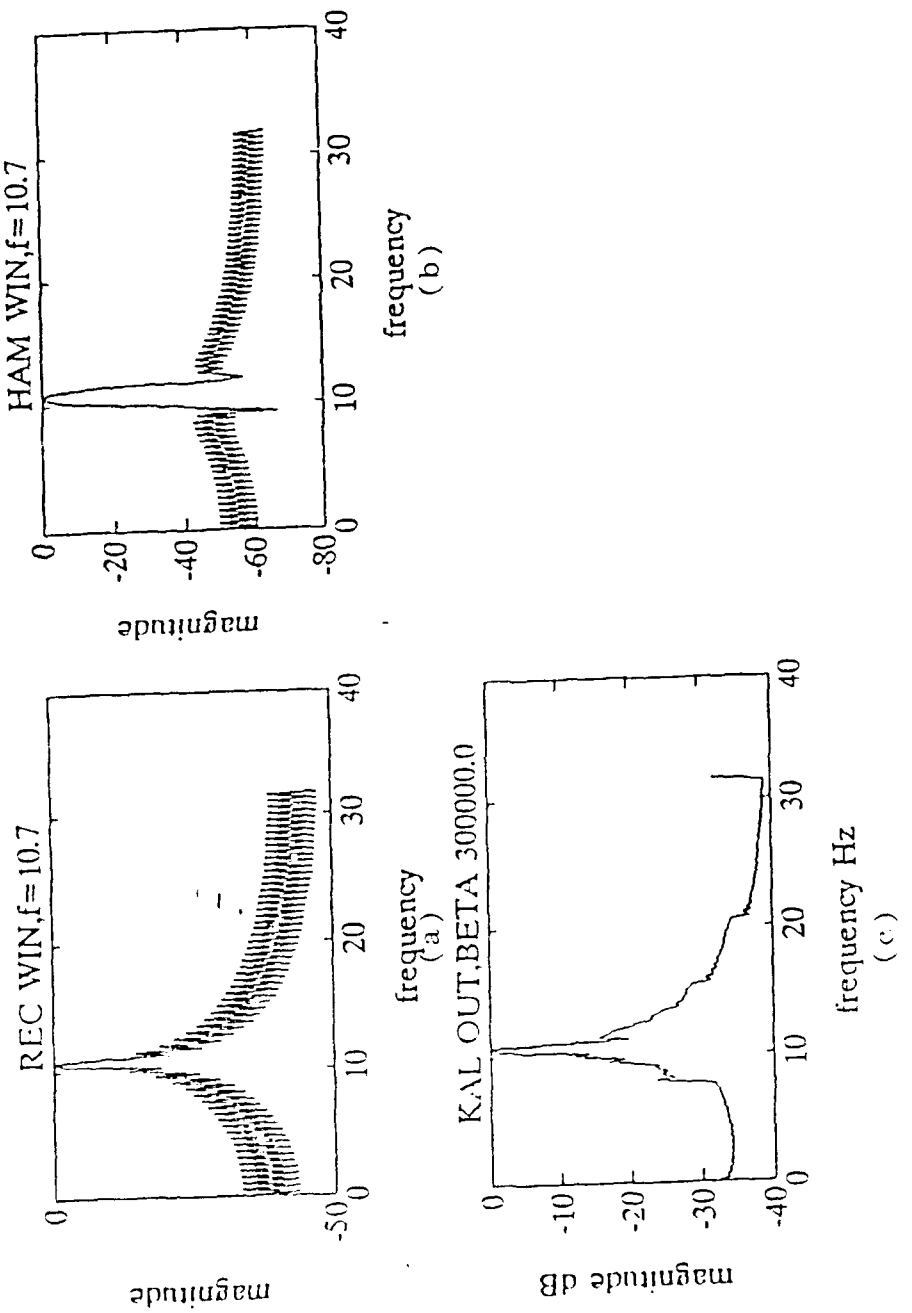


Figure 26. Signal-Only, Sinusoid (f = 10.7 Hz)
(a) Periodogram (rectangular window)
(b) Periodogram (Hamming window)
(c) Kalman Filter Output β = 300000

E. THE EFFECT OF DATA RECORD AND TRANSFORM LENGTH

Finally, the number of data points was increased from 128 to 256, 512 and 1024. The objective was to evaluate the performance of the Kalman filter for a given input signal strength (in this case +12 dB) at different length periodograms. In each case, the data record was zero-padded to twice its original length (i.e., 512 points zero-padded to 1024). Also in each case, the input SNR was decreased in order to compensate for the increased processing gain caused by the data record. Processing gain is approximated by:

$$\text{processing gain} \quad [\log_2 (\text{data record length})-1]\text{x}3 \text{ dB} \qquad (3.27)$$

For example, in our baseline case of 128 points, the expected processing gain is $[\log_2 (128)-1]\text{x}3$ dB = 18 dB. For an input SNR of -6 dB, the expected output SNR is then 18-6 = 12 dB, which is approximately the strength of the peak in Figure 17. For the longer data trials, the additive noise variance (power) was increased in order to maintain output SNR at approximately 12 dB. Initial results indicate a dependence of β on data/transform length. As the data/transform length increases, better results may be obtained by increasing β (see Appendix D).

IV. CONCLUSIONS

The Kalman filter can enhance the spectral peaks of a periodogram of an unwindowed time series. This is most apparent in the single spectral peak case. In the case of multiple spectral peaks, the resolution of the unfiltered periodogram is largely preserved since the Kalman filter will smooth the spectral estimate without major broadening the narrow band components. Using a filter parameter in the range 100,000 to 700,000 and a 128-point data record zero-padded to 256 points, reliable signal detection was achieved at SNR's of -6 dB of the time series. Signal detection is possible down to -12 dB (of the time series SNR), depending on the noise realization used.

Topics for further study are the application of the Kalman filter to multidimensional (time varying) spectra, and quantification of selection criteria for the filter parameter β. In addition, the dependence of β on input SNR, output SNR, record length and/or transform length should be examined. Another possible follow-on project is the development of an enhanced Kalman filtering algorithm that adjusts the parameter β based on the assignment of signal or noise only. This would mean faster filter response during signal portions and slower response during noise-only segments.

APPENDIX A
COMPUTER CODE

The Kalman filtering program was originally written in FORTRAN 77. The FORTRAN code is given in Appendix A.1. For this thesis, the filter program was converted to PC-MATLAB (Version 3.13) and simulations run on an 80386-based IBM compatible PC. The MATLAB code for the filtering program is given in Appendix A.2

SECTION A.1
FORTRAN Computer Code

```
c       ** Nonstationary filtering using
c       ** suboptimal kalman filtering (local
c       ** average), and gibbs field with
c       ** anihiling.
c
c       ** The  input file must given on INPUT.DAT
c       ** The filtered output file is stored on OUTPUT.DAT
c       ** The detected breackpoints are given in MODEL.DAT
c       ** All these *.DAT files are ASCII.
c
c       ** The program now works for 128 data points (see the variable
c       ** "npoints" below.  This can be changed to any number of points.
c
c       ** The program requires to enter 2 parameters:
c       ** "sigma":  the value of the noise standard deviation (nonzero);
c       ** "beta ":  a positive parameter. It is a measure of the probability
c       **           the signal having a jump. As is now this parameter is
c       **           set by pure trial and error.  If you get too many
c       **           jumps detected it means that beta is too low. If you get
c       **           too few jumps it means that beta is too large. However
c       **           usually the best value of beta depends on the signal to
c       **           noise ratio of the data.
c
        real yin(256), y(256), x(2,256)
        real k1,k2
        integer pointer(2, 256), t, out
        integer mout(256)
        open(1, file='output.dat', status='old')
        open (2, file= 'input.dat', status='old')
        open(3, file= 'model.dat', status='old')
c       ** get data from file
        rewind 1
        rewind 2
        rewind 3
c       **
        npoints=128
c       **
        do 50 t=1,npoints
        read (2,222) y(t)
        yin(t)=y(t)
c       write(*, 111) y(t)
50      continue
222     format(f8.4)
c       ** enter data and initialize
        write(*,555)
555     format('  ENTER: sigma,beta  (>0)')
        read(*,666) sigma, beta
666     format(2f10.4)
        sv2=sigma**2
c
        e1=0.0
        e2=0.0
        d1=0.0
        d2=0.0
        x(1,1)=y(1)
        x(2,1)=y(1)
        tau=1
c
c       ** main loop
        do 100 t=1, (npoints-1)
```

```
c
      k1=1.0/(tau+1.0)
      x11=x(1,t) + k1*(y(t)-x(1,t))
      d11=d1-beta
      e11=e1+(1.0/(2.0*sv2))*(y(t+1)-x11)**2
      c11=e11+d11
c
      k2=1.0
      x12=x(1,t) + k2*(y(t)-x(1,t))
      d12=d1+beta
      e12=e1+(1.0/(2.0*sv2))*(y(t+1)-x12)**2
      c12=e12+d12
c
      k1=0.5
      x21=x(2,t) + k1*(y(t)-x(2,t))
      d21=d2-beta
      e21=e2+(1.0/(2.0*sv2))*(y(t+1)-x21)**2
      c21=e21+d21
c
      k2=1.0
      x22=x(2,t) + k2*(y(t)-x(2,t))
      d22=d2+beta
      e22=e2+(1.0/(2.0*sv2))*(y(t+1)-x22)**2
      c22=e22+d22
c
c     write(*,444) t,d11,e11,d12,e12,d21,e21
444   format(i3,3(f10.2,f10.2,2x))
c
c     *** update states in dynamic prog.
      if(c11.le.c21) then
              x(1,t+1)=x11
              e1=e11
              d1=d11
              c1=e1+d1
              pointer(1,t+1)=1
              tau=tau+1
      else
              x(1,t+1)=x21
              e1=e21
              d1=d21
              c1=e1+d1
              tau=2
              pointer(1,t+1)=2
      endif
c
      if (c22.lt.c12) then
              x(2,t+1)=x22
              e2=e22
              d2=d22
              c2=e2+d2
              pointer(2,t+1)=2
      else
              x(2,t+1)=x12
              e2=e12
              d2=d12
              c2=e2+d2
              pointer(2,t+1)=1
      endif
c
c
```

```
100     continue
c
c       backward substitution and smoothing
        tau=1.0
        if(c1.le.c2) then
                out=1
        else
                out=2
        endif
c
        y(npoints)=x(out,npoints)
        n2=0
        do 150 t=npoints,2,-1
                out=pointer(out,t)
                xout=x(out,t-1)
        if (out.eq.2) then
                tau=1.0
        else
                tau=tau+1.0
        endif
c
        y(t-1)=y(t)+(1.0/tau)*(xout-y(t))
        mout(t-1)=out*100
c       write(1,111) xout
        n2=n2+out-1
c       write(*,333) t, out
333     format(2(2x,i5))
150     continue
c
        sigma=0.0
        do 800 t=1,npoints
        write(1,111) y(t)
        write(3,334) mout(t)
334     format(i5)
        ye=(y(t)-yin(t))**2
        sigma=sigma + (1.0/t)*(ye-sigma)
800     continue
        sigma=sqrt(sigma)
        write(*,777) sigma, n2, npoints
777     format('   sigma=', f8.4, '  n2=',I5,'   npoints=',I5)
111     format( f8.4)
c
        rewind 1
        rewind 2
        stop
        end
```

SECTION A.2
PC-MATLAB Computer Code

```
%   EC THESIS                                                GO, W. W.

%   THE10.M  KALMAN FILTER APPLIED TO PERIODOGRAM OF TWO SINUSOIDS
%            IN GAUSSIAN WHITE NOISE. 128 DATA POINTS ARE ZERO-
%            PADDED TO 256 AND THEN THE PERIODOGRAM IS COMPUTED.
%            ONLY HALF OF THE RESULTING FREQUENCY POINTS (UP TO
%            ONE-HALF OF THE SAMPLING FREQUENCY) ARE PLOTTED AND USED
%            AS INPUT TO THE KALMAN FILTER. THE FOLLOWING CASES
%            ARE PLOTTED:
               1) PERIODOGRAM, RECTANGULAR WINDOW ON TIME DATA
               2) PERIODOCRAM, HAMMING WINDOW ON TIME DATA
               3) OUTPUT OF KALMAN FILTER APPLIED TO PERIODOGRAM
                  OF RECTANGULARLY WINDOWED DATA (CASE 1).

%
%    The program requires 2 parameters to be specified:
%     "sigma":  the value of the noise standard deviation (nonzero);
%     "beta ":  A positive parameter. It is a measure of the probability
%               of the signal having a jump. Now this parameter is
%               set by pure trial and error.  If you get too many
%               jumps detected it means that beta is too low. If you get
%               too few jumps it means that beta is too large.

%   NOTE 1: THIS PROGRAM UTILIZES MATLAB FUNCTIONS PER.M AND PERLN.M
%           (CODE FOLLOWS MAIN PROGRAM) TO COMPUTE THE PERIODOGRAM
%           IN dB AND LINEAR UNITS RESPECTIVELY.

%   NOTE 2: THIS PROGRAM UTILIZES MATLAB FUNCTION FVEC.M (CODE
%           FOLLOWS MAIN PROGRAM) TO CREATE A FREQUENCY VECTOR
%           FOR PLOTTING.


%      f1= 10.7Hz, NOT A BIN CENTER
%      f2= 11.2, NOT A BIN CENTER

%      fs = 64 Hz, SAMPLING FREQUENCY
%      128 DATA POINTS ZERO-PADDED TO 256
%      WHITE NOISE VARIANCE = 4000/2000
%      INPUT SNR -6.02dB

clear
clg
f1= 10.7  ;                 % f is frequency
f2= 11.2  ;
fs = 64   ;                 % fs is sampling frequency

nvar=4000/2000;             % noise variance

for n= 0 : 127 ;            % compute signal vector
      x(n+1) = cos(n*2*pi*(f1/fs)) + cos(n*2*pi*(f2/fs));
end


rand('normal');
rand('seed',3 )
nz=sqrt(nvar).*rand(1:128);       % noise vector

xn=x +nz;                         % corrupt signal with noise
```

```
w=hamming(128);                    % Hamming window
xnw=w'.*xn;                        % apply Hamming window


xp= [ xn  zeros(1:128) ];

xpw= [ xnw  zeros(1:128) ];

psd =perln(xp);                    % periodogram(linear units)
test=per(xp);                      % periodogram(dB)

testw=per(xpw);

freq=fvec(64,xp);                  % frequency vector for plotting

subplot(211),plot(freq(1:128),test(1:128))
title('THE10:2 SIN IN NOISE,SNR -6.02dB,f1=10.7,f2=11.2')
xlabel('frequency')
ylabel('magnitude')

%   subplot(211),plot(freq(1:128),testw(1:128))
%   title('THE10:2 SIN IN NOISE,HAM WIN,SNR -6.02dB,f1=10.7,f2=11.2')
%   xlabel('frequency')
%   ylabel('magnitude')
%   meta preplt2
%   pause


y= psd(1:128);


%       KALMAN FILTER
%       y IS DATA RECORD. FILTER IS APPLIED TO PERIODOGRAM IN
%       LINEAR UNITS.

        x=zeros(2,128);
        pointer=zeros(2,128);
        yin = y;

        beta =500000.0;                % filter parameter,
        sigma = sqrt(nvar);            % noise standard deviation
        sv2=sigma^2;

        npoints=length(y);

        e1=0.0;
        e2=0.0;

        d1=0.0;
        d2=0.0;

        x(1,1)=y(1);
        x(2,1)=y(1);

        tau=1.0;

%       MAIN LOOP
```

```
for t=1:(npoints-1);
        k1=1.0/(tau+1.0);
        x11=x(1,t)+k1*(y(t)-x(1,t));
        d11=d1-beta;
        e11=e1+(1.0/(2.0*sv2))*((y(t+1)-x11)^2);
        c11=e11+d11;

        k2=1.0;
        x12=x(1,t)+k2*(y(t)-x(1,t));
        d12=d1+beta;
        e12=e1+(1.0/(2.0*sv2))*((y(t+1)-x12)^2);
        c12=e12+d12;

        k1=0.5;
        x21=x(2,t)+k1*(y(t)-x(2,t));
        d21=d2-beta;
        e21 = e2+(1.0/(2.0*sv2))*((y(t+1)-x21)^2);
        c21=e21+d21;

        k2=1.0;
        x22=x(2,t)+k2*(y(t)-x(2,t));
        d22=d2+beta;
        e22=e2+(1.0/(2.0*sv2))*((y(t+1)-x22)^2);
        c22=e22+d22;

%       UPDATE STATES IN DYNAMIC PROGRAM.

          if c11<c21
               x(1,t+1)=x11;
               e1=e11 ;
               d1=d11 ;
               c1=e1+d1 ;
               pointer(1,t+1)=1    ;
               tau = tau+1;

           else
               x(1,t+1)=x21   ;
               e1=e21;
               d1=d21;
               c1=e1+d1;
               tau=2;
               pointer(1,t+1)=2;

           end

           if c22<c12
                x(2,t+1)=x22;
                e2=e22;
                d2=d22;
                c2=e2+d2;
                pointer(2,t+1)=2;

           else
                x(2,t+1)=x12;
                e2=e12;
                d2=d12;
                c2=e2+d2;
                pointer(2,t+1)=1;
```

```
                end

        end

%       END MAIN LOOP

%       BACKWARDS SMOOTHING AND SUBSTITUTION

        tau=1.0;

        if c1<c2
                out=1;
        else
                out=2;
        end

        y(npoints)=x(out,npoints);

        for t=128:-1:2
                  out=pointer(out,t);
                  xout=x(out,t-1);

                  if out==2
                           tau=1.0;
                           y(t-1)=xout;
                  else
                           tau=tau+1;
%                 end

%             y(t-1)=y(t)+(1.0/tau)*(xout-y(t));
              y(t-1)=xout;
              end


              trans(t-1)=out;
         end

        ynorm=(1/max(y)).*y;
        ydb=10*log10(ynorm);

        ysh= [ ydb(2:length(ydb))  ydb(1) ];

        subplot(212),plot(freq(1:128),ysh)
        title('THE10:KAL,SNR -6.02,BETA 500000.0')

        xlabel('frequency')
        ylabel('magnitude dB')

         meta preplt3
%        pause

%       plot(trans,'+'),title('transition pts')
```

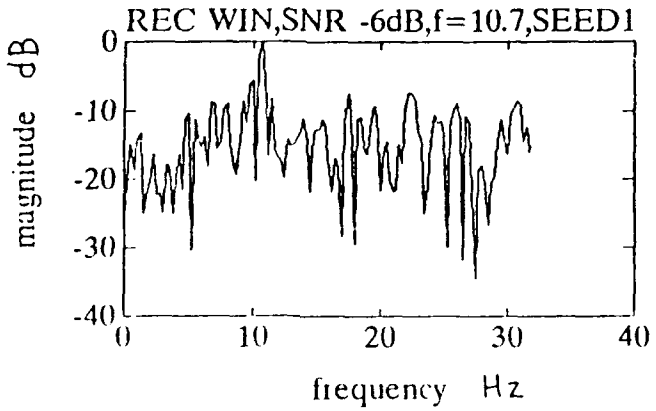
```
%       EC THESIS                                          GO. W.W.
%       PER.M    COMPUTE THE PERIODOGRAM OF DATA VECTOR X

function y=per(x)
l=length(x);
tr=fft(x);

for i=0:(l-1);
    ps(i+1)=(abs(tr(i+1)))^2;
end

psnorm=(1/max(ps)).*ps;
y=10*log10(psnorm);
```

```
%       EC THESIS                                          GO. W.W.

%       PERLN.M     COMPUTE THE PERIODOGRAM OF DATA VECTOR X
%                   LINEAR UNITS

        function y=perln(x)
        l=length(x);
        tr=fft(x);

        for i=0:(l-1);
            ps(i+1)=(abs(tr(i+1)))^2;
        end

        y=ps;
```

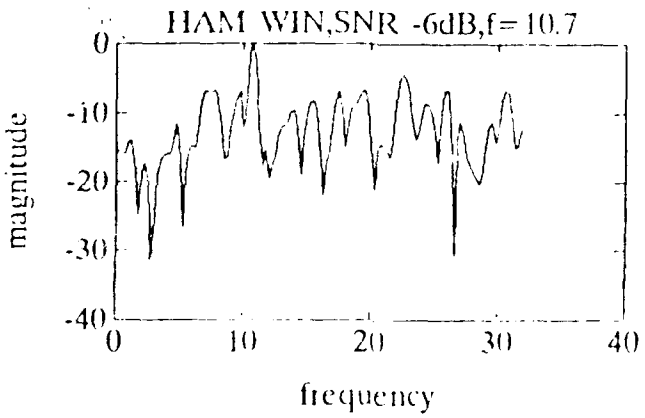
```
%       EC THESIS                                          GO,W.W.

%       FVEC.M          CREATE THE FREQUENCY VECTOR USED IN PLOTTING
%                       A PERIODOGRAM. fs IS THE SAMPLING FREQUENCY
%                       AND X IS THE DATA VECTOR.

        function f=fvec(fs,x)

        n=length(x);
        f=fs*(0:n-1)/n;
```

APPENDIX B
EFFECTS OF THE KALMAN FILTER PARAMETER

The effects of changing the parameter β on the performance of the Kalman filter were investigated. The test data was a single sinusoid, with frequency of 10.7 Hz, embedded in Gaussian white noise. The frequency 10.7 Hz was specifically chosen so as not to be at a bin center of the FFT. A record of 128 data points was zero-padded to 256. The input SNR of the time series was -6 dB. The same noise realization was used for all runs. Figure B.1 shows the unfiltered (rectangular windowed) and Hamming windowed periodograms. Figures B.2a through B.2j show the filtered periodograms for ten different values of β. As discussed in Chapter 3, β in the range 100,000 to 700,000 produced the best results. Values for β below this range tend not to smooth the spectral estimate enough to significantly enhance the main spectral peaks. Values of β above this range tend to oversmooth and obliterate the spectral estimate (depending on the noise realization).

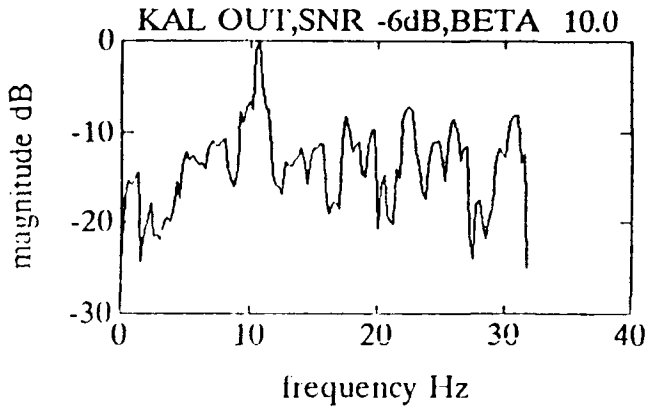


(a) Rectangular window

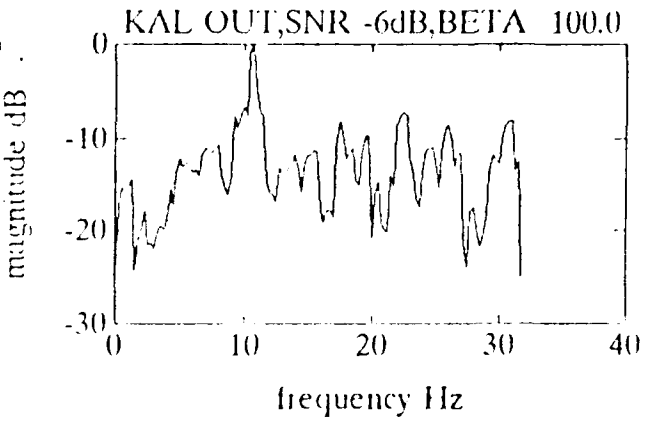


(b) Periodogram (Hamming window)

Figure B.1. Unfiltered Periodograms

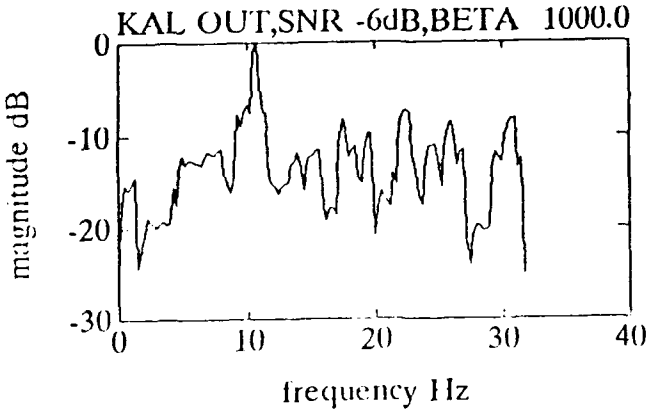


(a) β = 10.0

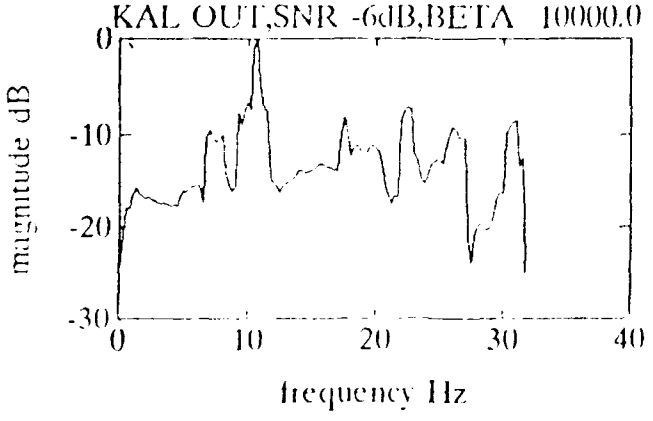


(b) β = 100.0

Figure B.2. Output of the Kalman Filter

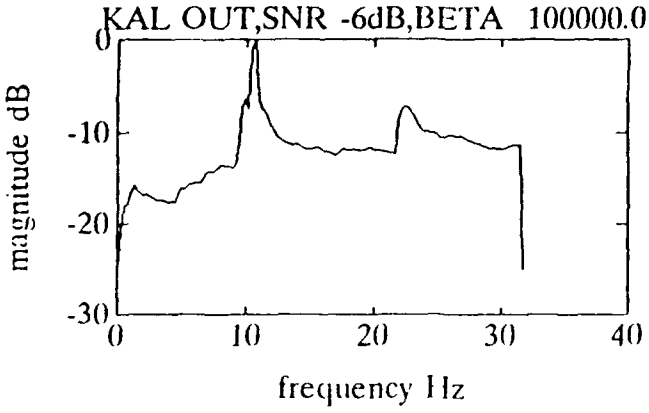


(c) β= 1,000.0

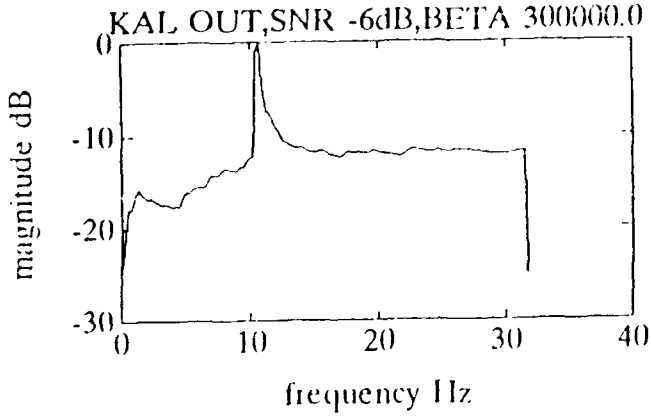


(d) β = 10,000.0

Figure B.2. Output of the Kalman Filter cont.

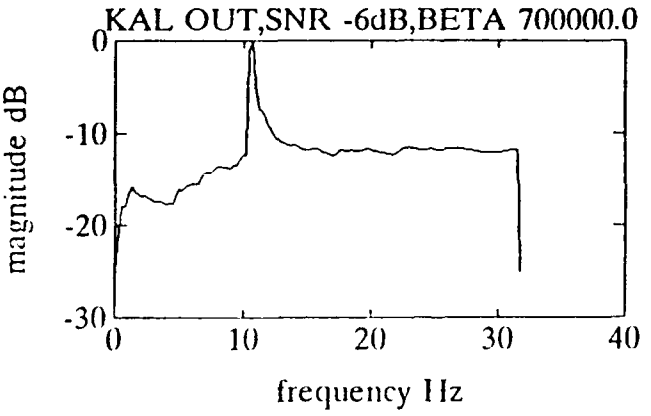


(e) β = 100,000.0

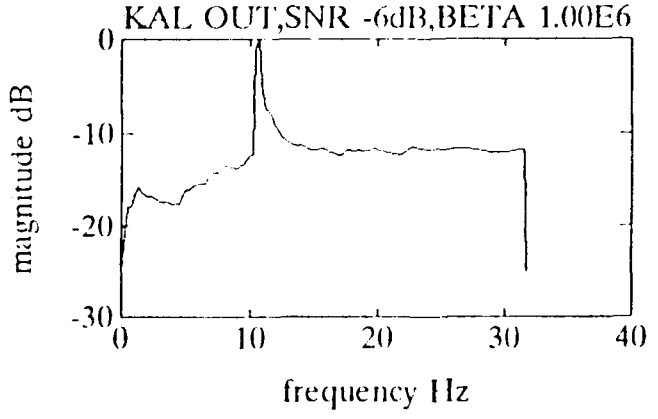


(f) β = 300,000.0

Figure B.2. Output of the Kalman Filter cont.

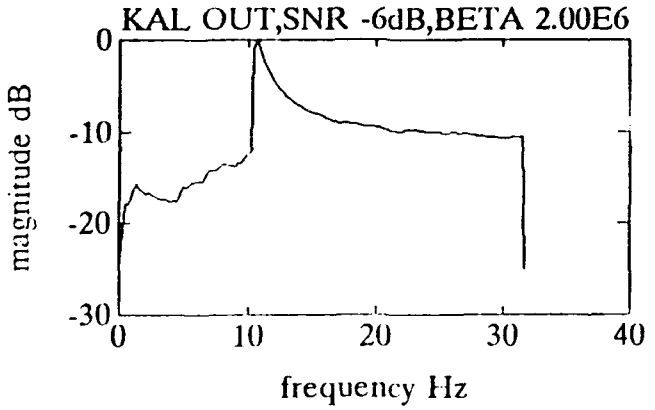


(g) β= 700,000.0

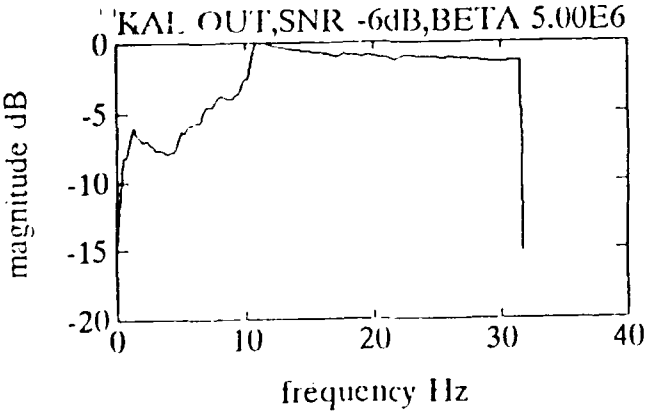


(h) β = 1.00 E6

Figure B.2. Output of the Kalman Filter cont.



(i) β = 2.00 E6



(j) β = 5.00 E6

Figure B.2. Output of the Kalman Filter cont.

APPENDIX C
PERFORMANCE OF THE KALMAN FILTER AT DIFFERENT INPUT SNR'S ON MULTIPLE NOISE REALIZATIONS

The performance of the Kalman filter at different input SNRs was evaluated. The test case was a single sinusoid, frequency 10.7 Hz (not a FFT bin center). A record of 128 data points was zero-padded to 256. The Kalman filter was run on data with time series SNRs of -3, -6, -9, and -12 dB. Ten different noise realizations were used at each SNR. Plots are shown in Figures C.1 through C.40. For comparison, the unfiltered and Hamming windowed pe_iodograms are also shown for each simulation. At -6 dB (time series SNR), reliable detection was achieved for all noise realizations tested.

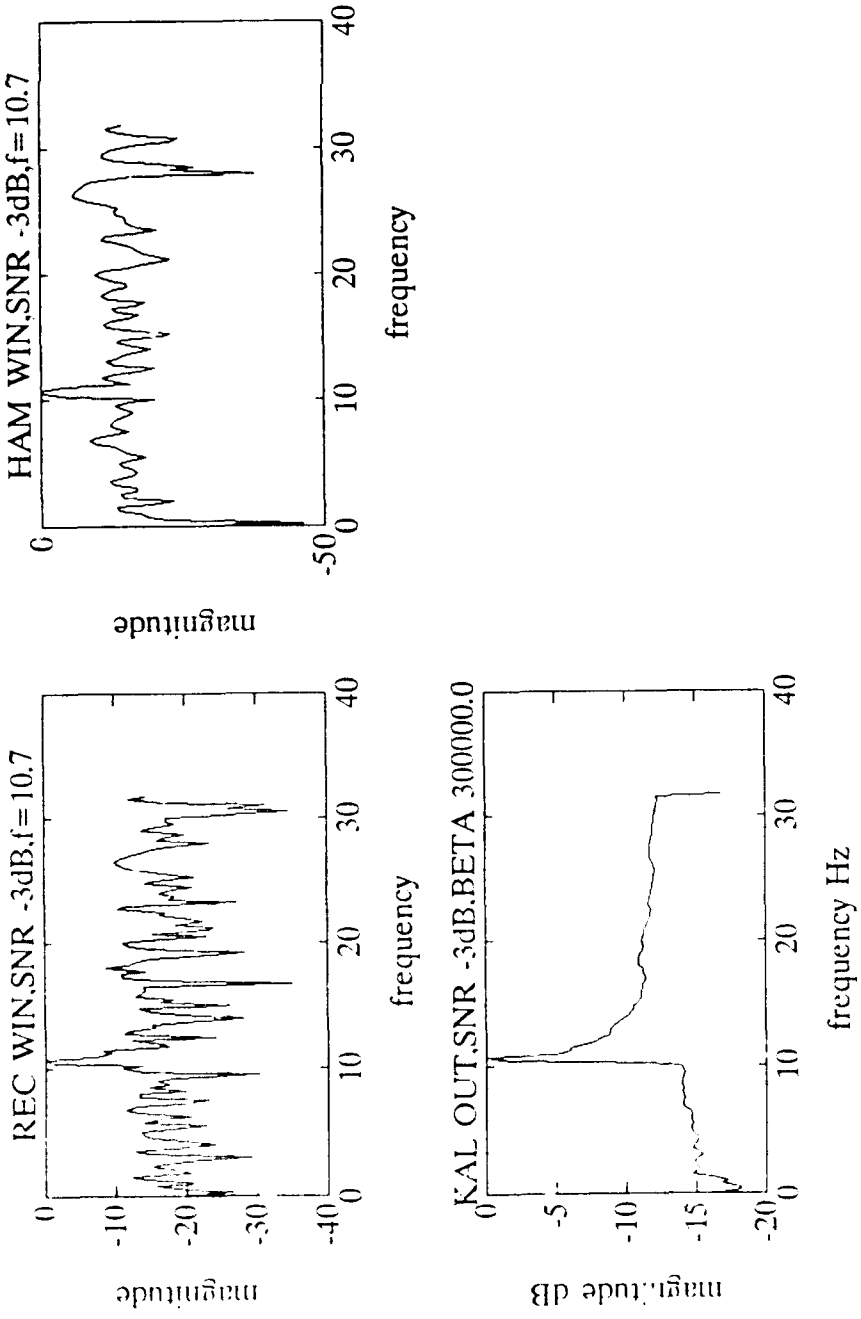


Figure C.1. -3 dB Input SNR, Noise Realization 1

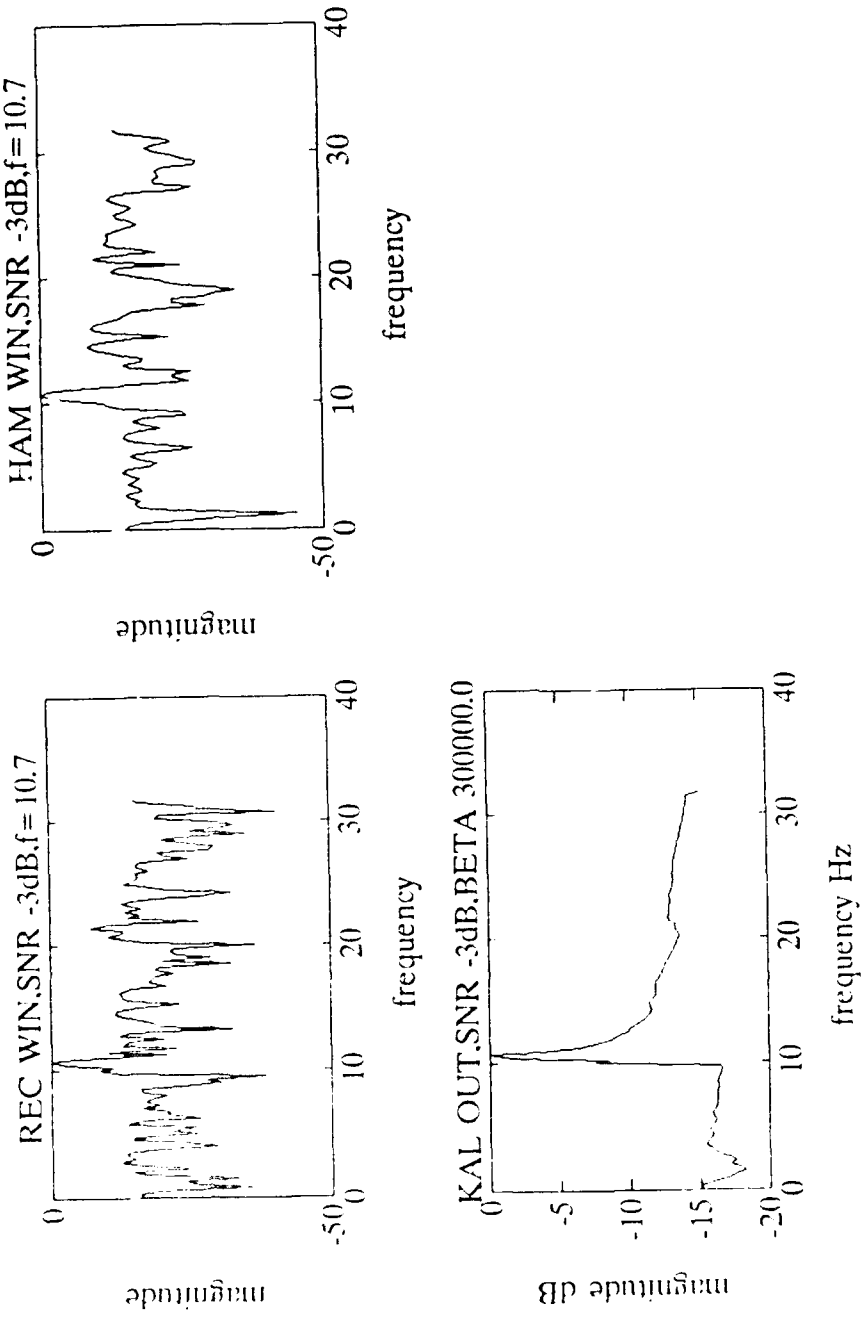


Figure C.2. -3 dB Input SNR, Noise Realization 2

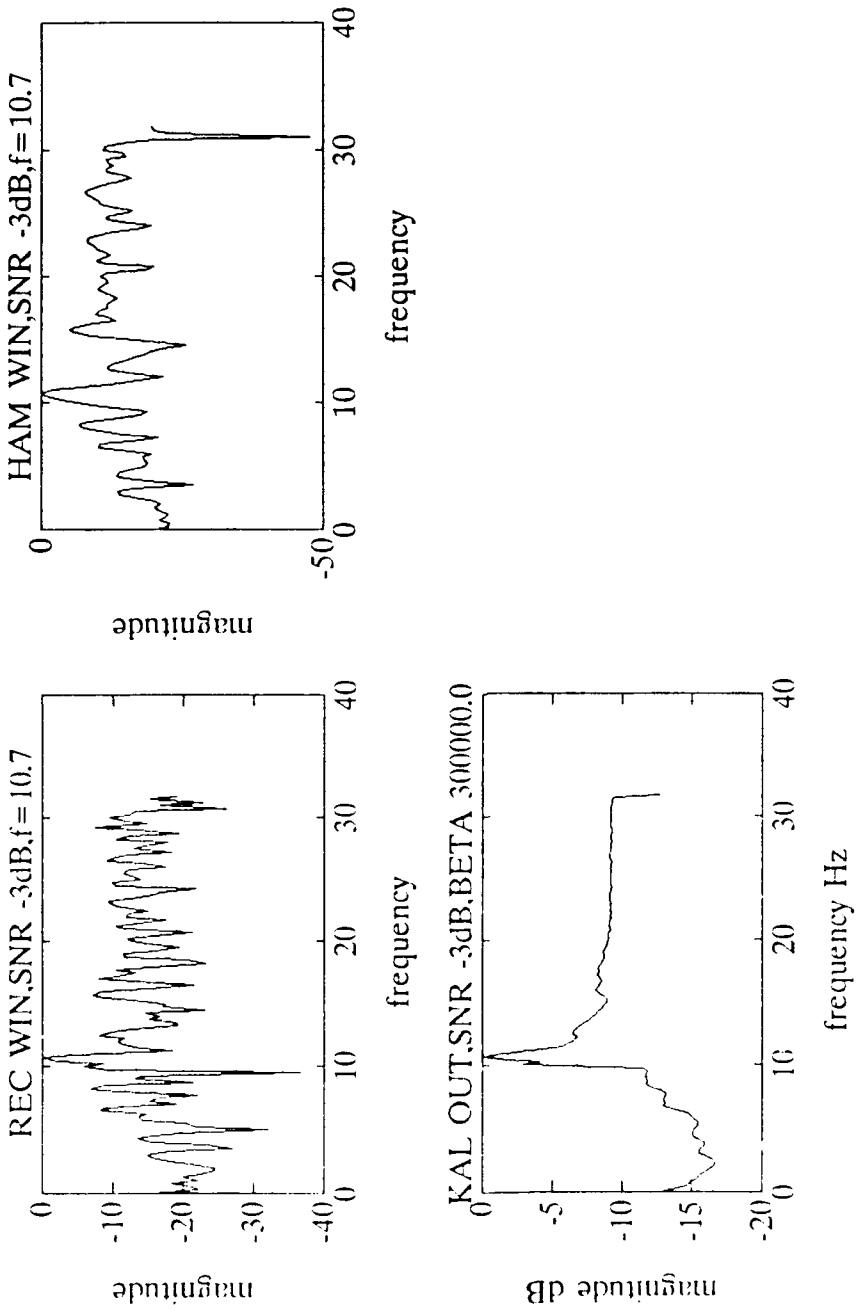


Figure C.3. -3 dB Input SNR, Noise Realization 3

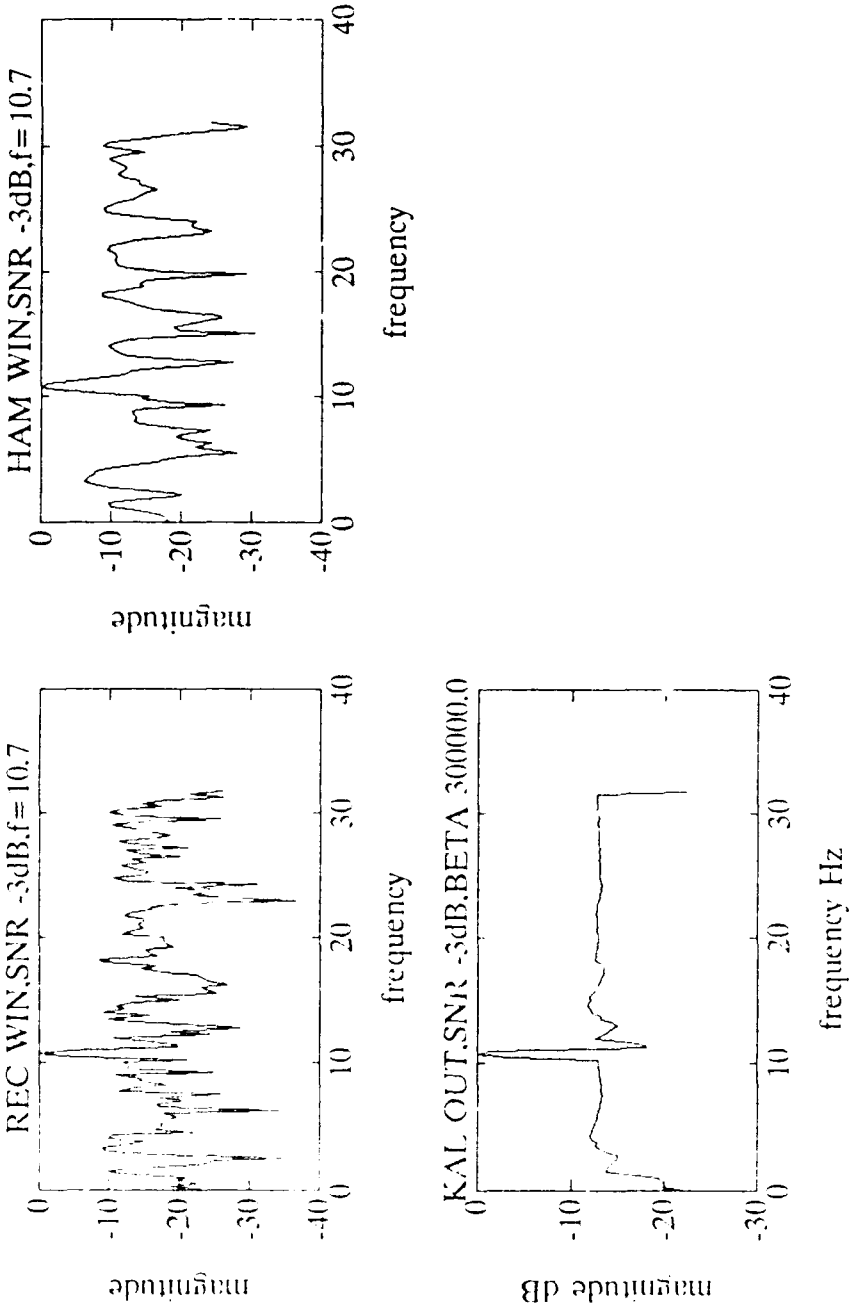


Figure C.4. -3 dB Input SNR, Noise Realization 4

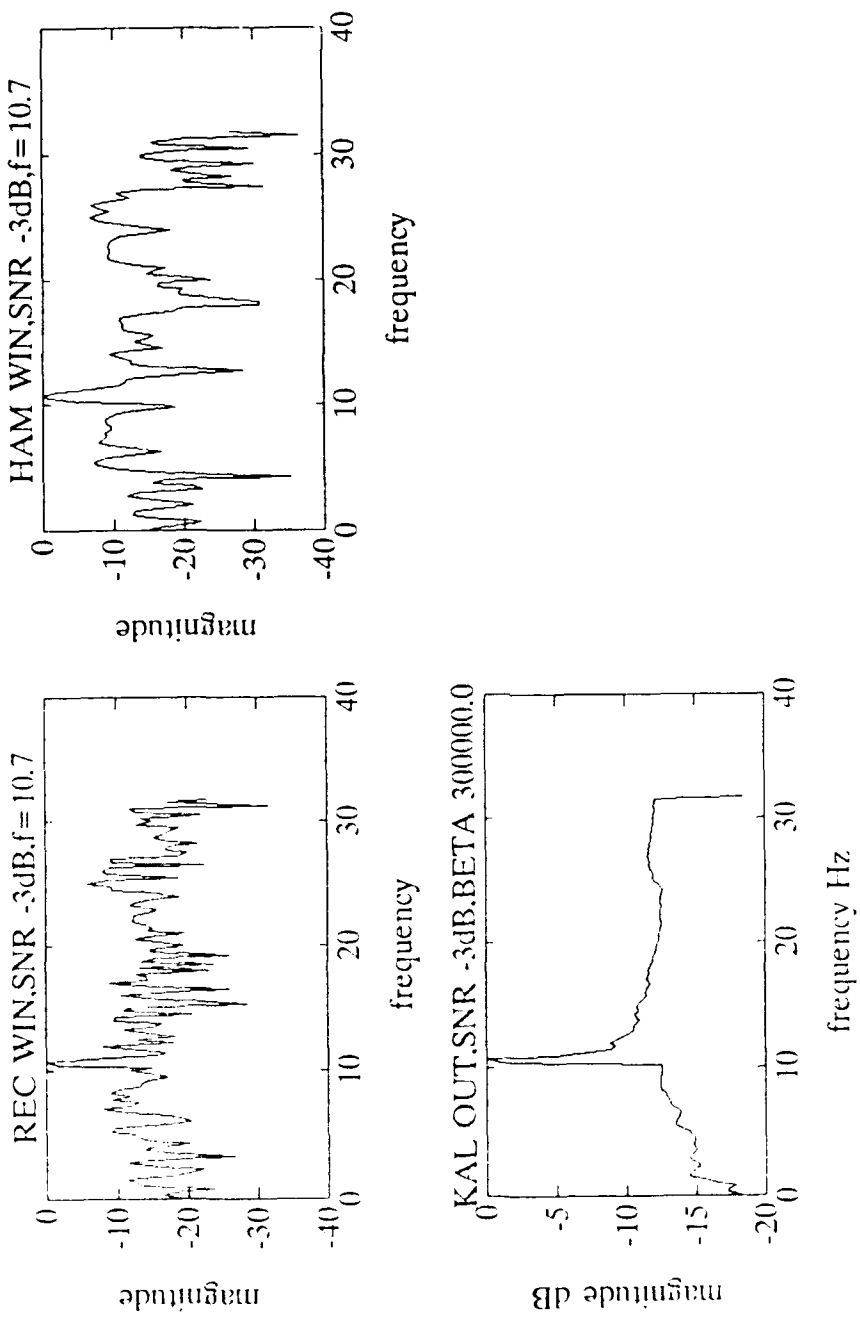


Figure C.5. -3 dB Input SNR, Noise Realization 5

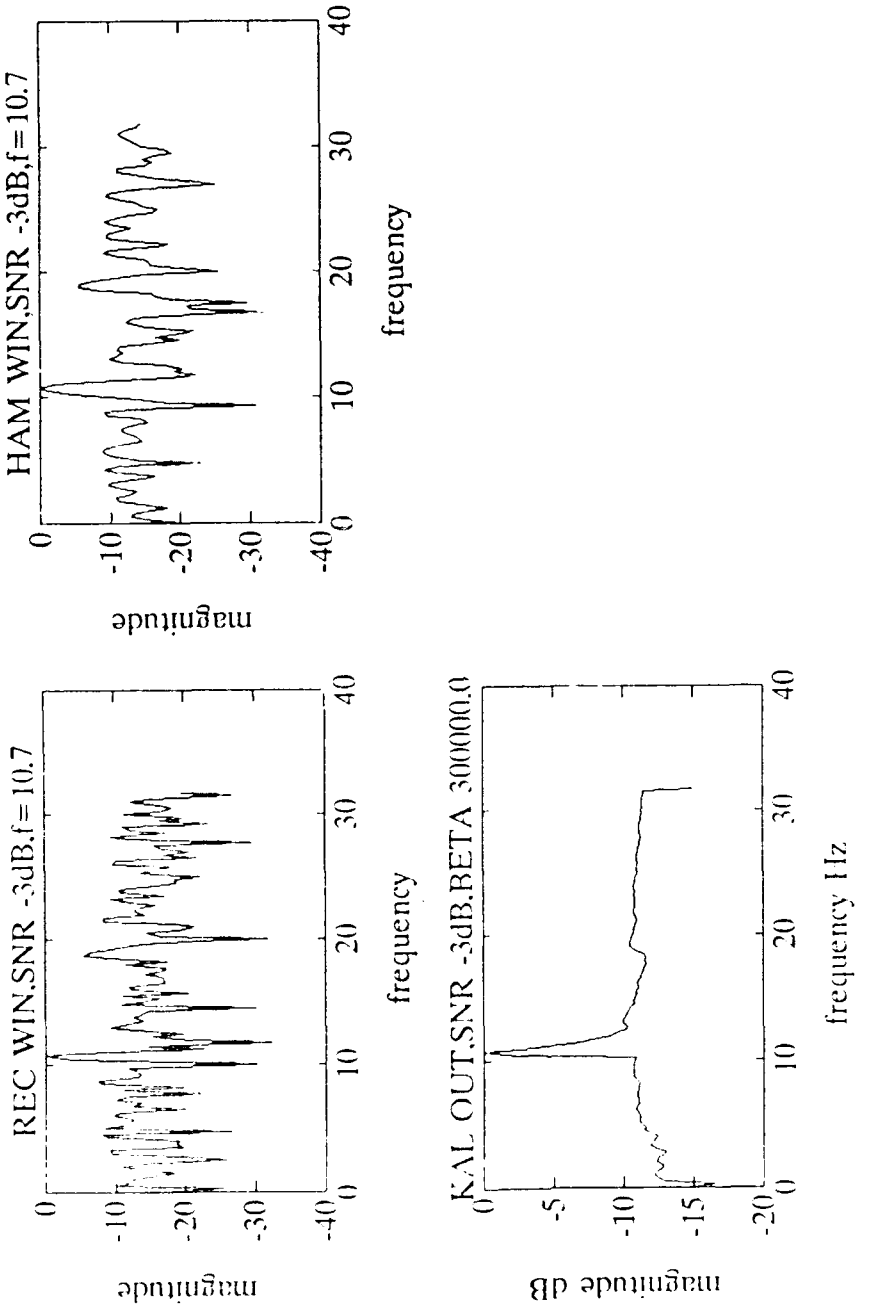


Figure C.6. -3 dB Input SNR, Noise Realization 6

Figure C.7. -3 dB Input SNR, Noise Realization 7

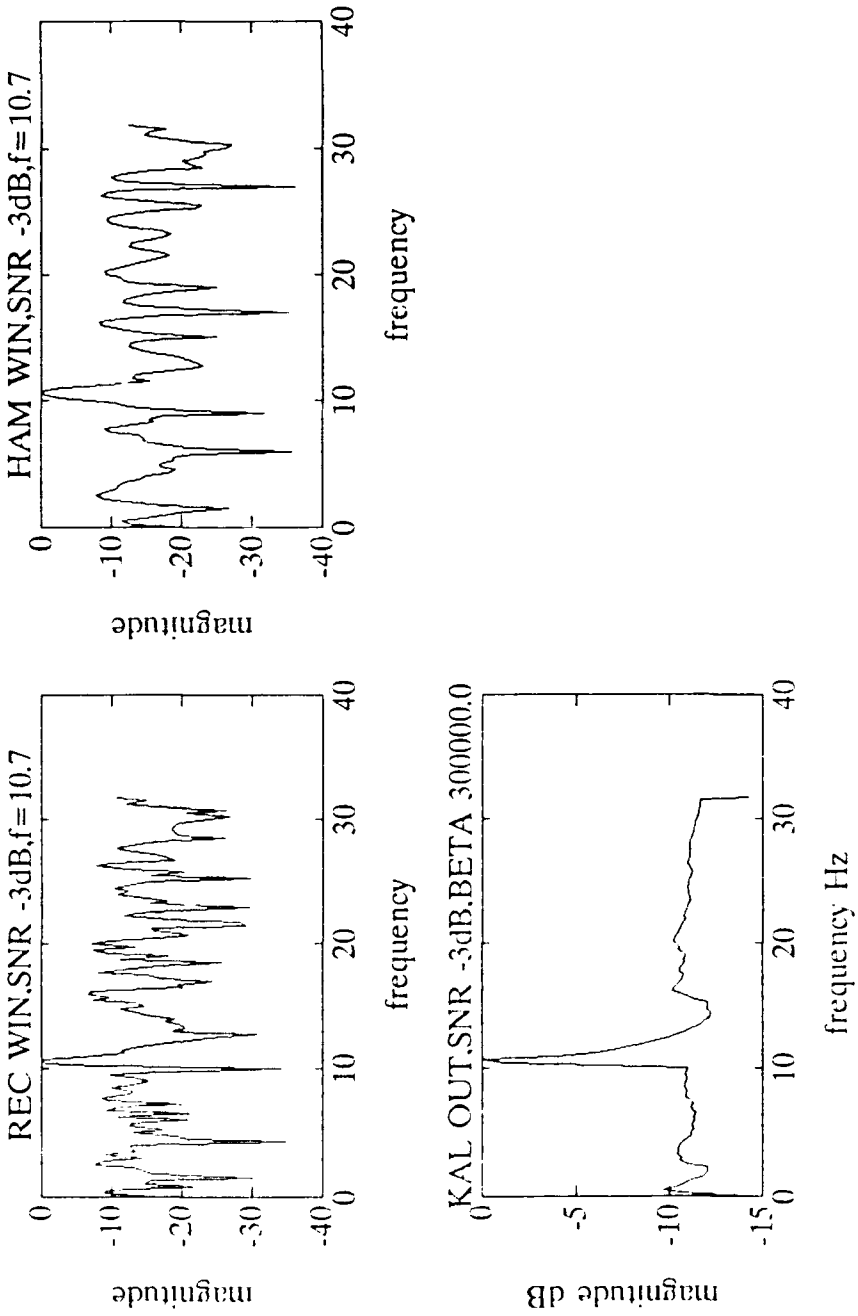


Figure C.9. -3 dB Input SNR, Noise Realization 9

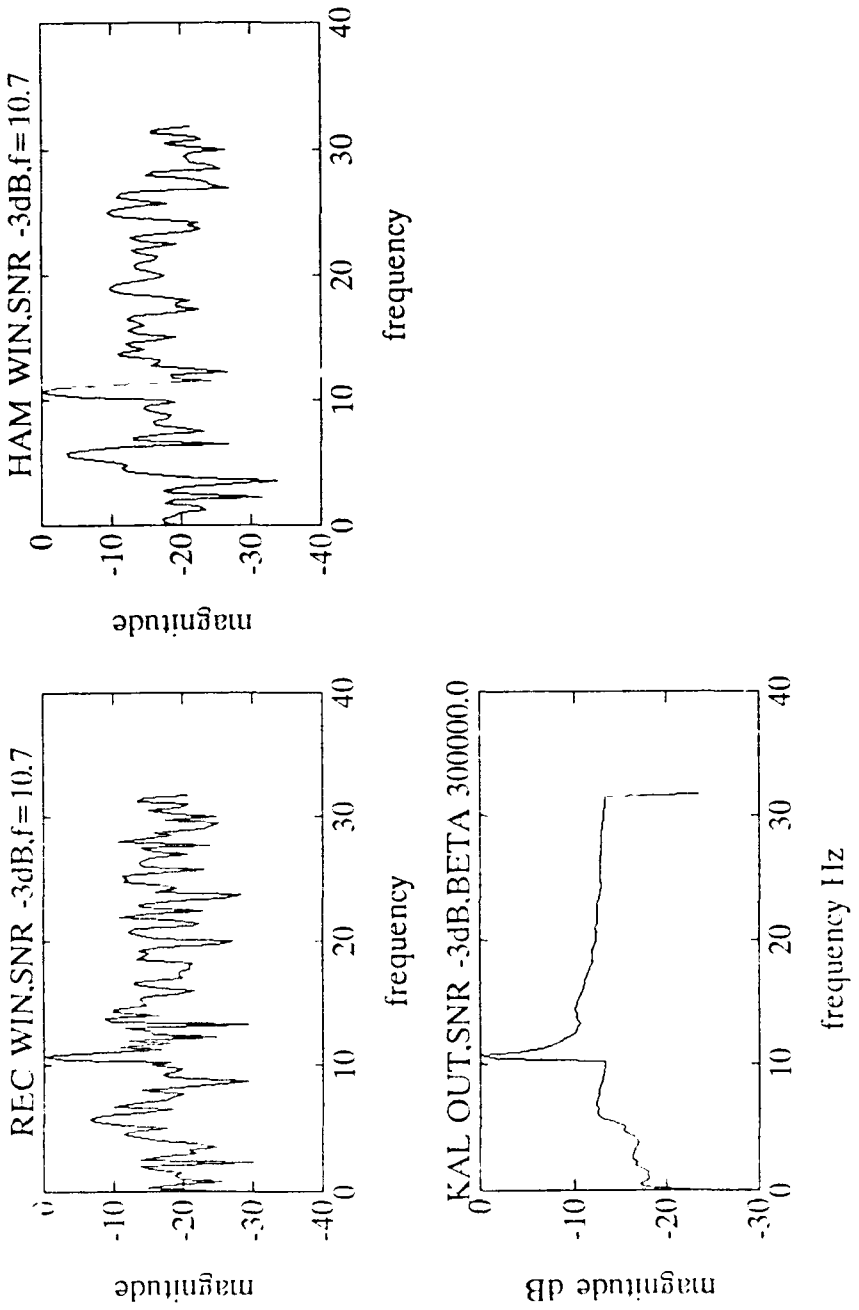


Figure C.10. -3 dB Input SNR, Noise Realization 10

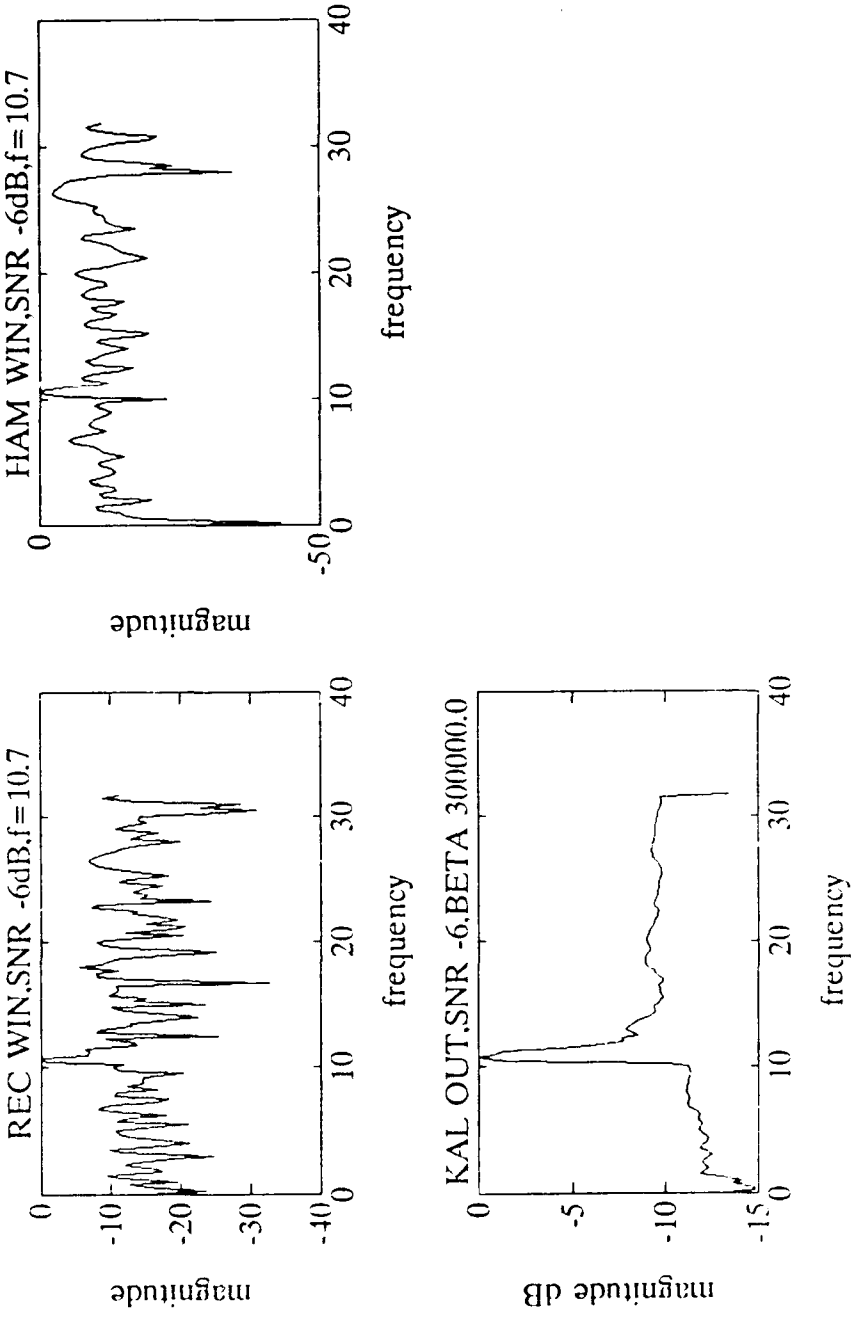


Figure C.11. -6 dB Input SNR, Noise Realization 1

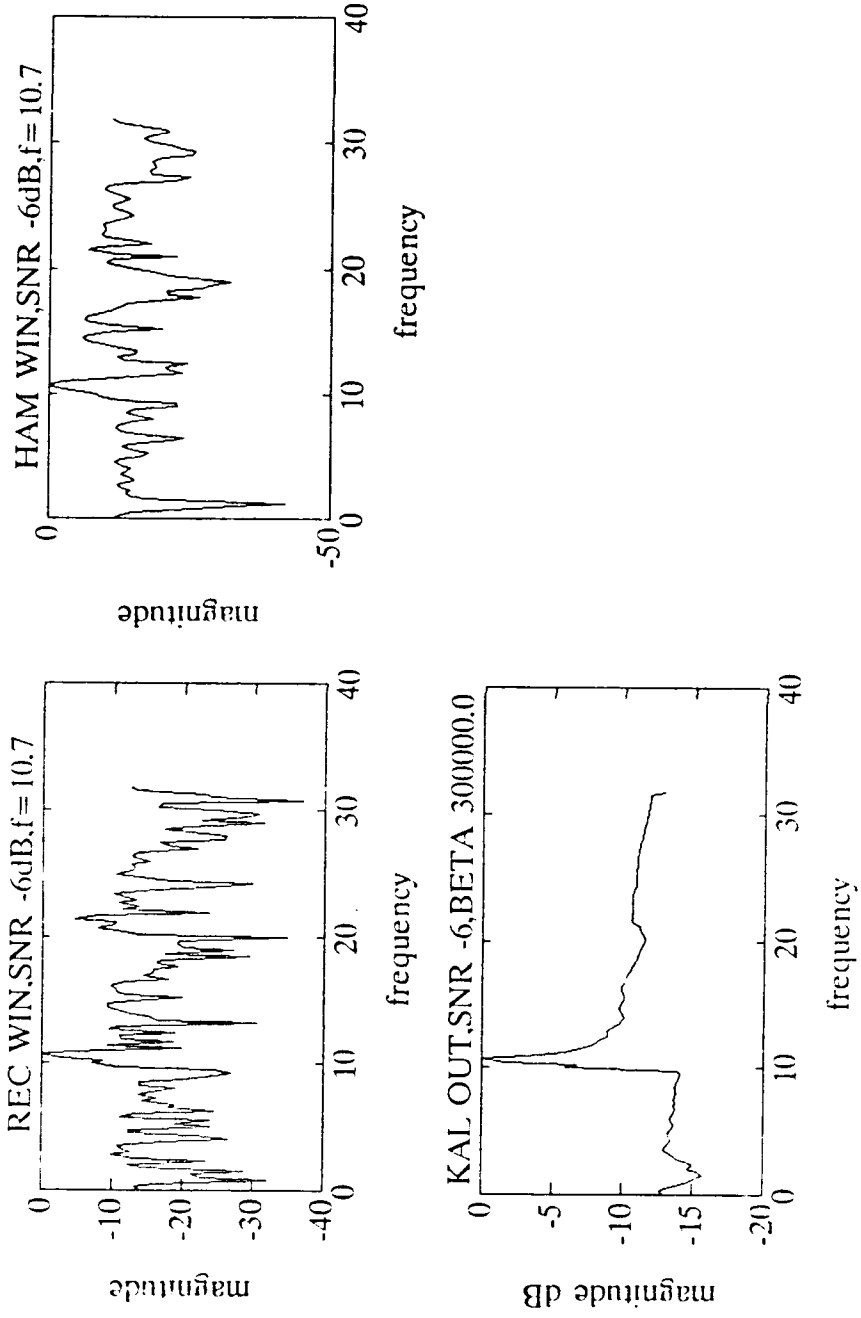


Figure C.12. -6 dB Input SNR, Noise Realization 2

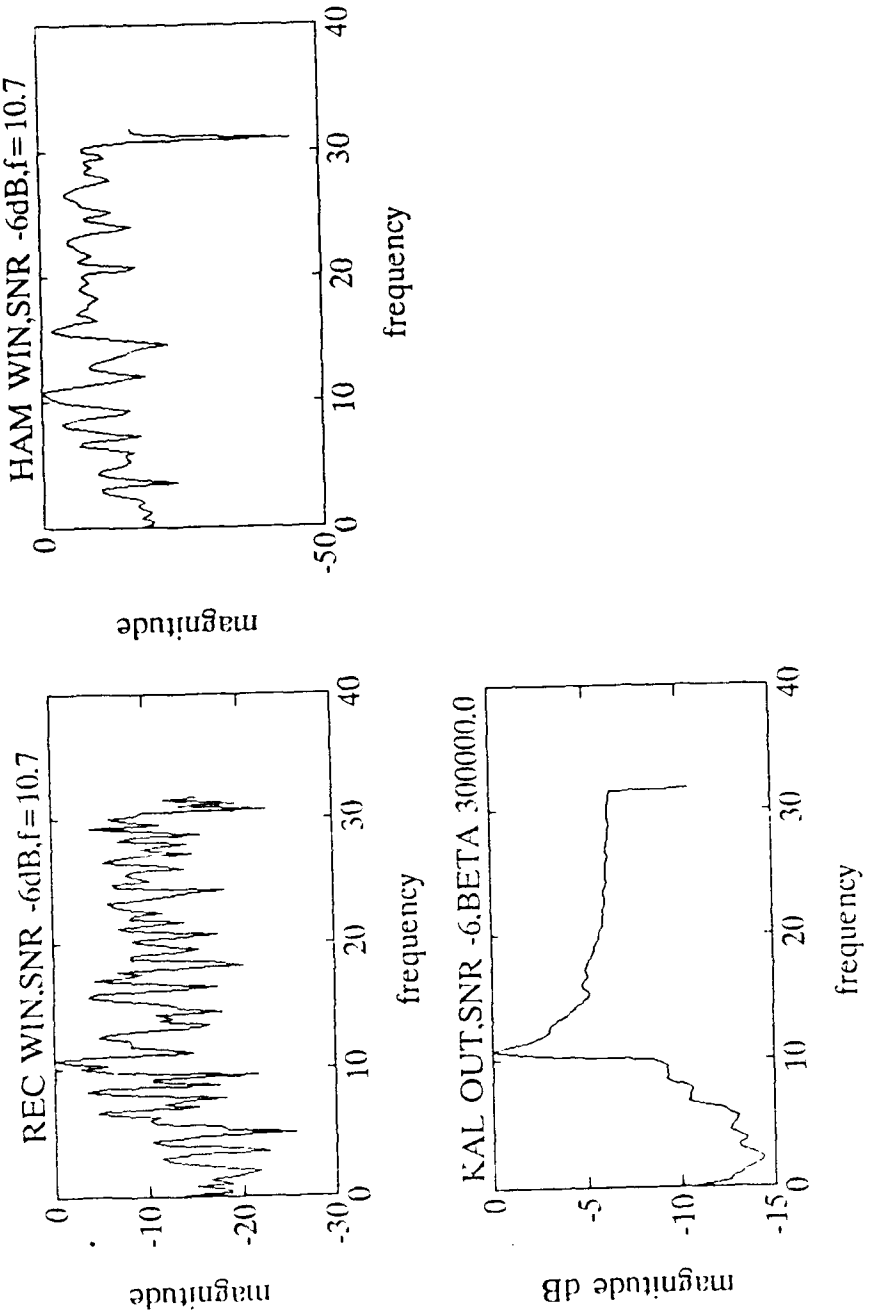


Figure C.13. -6 dB Input SNR, Noise Realization 3

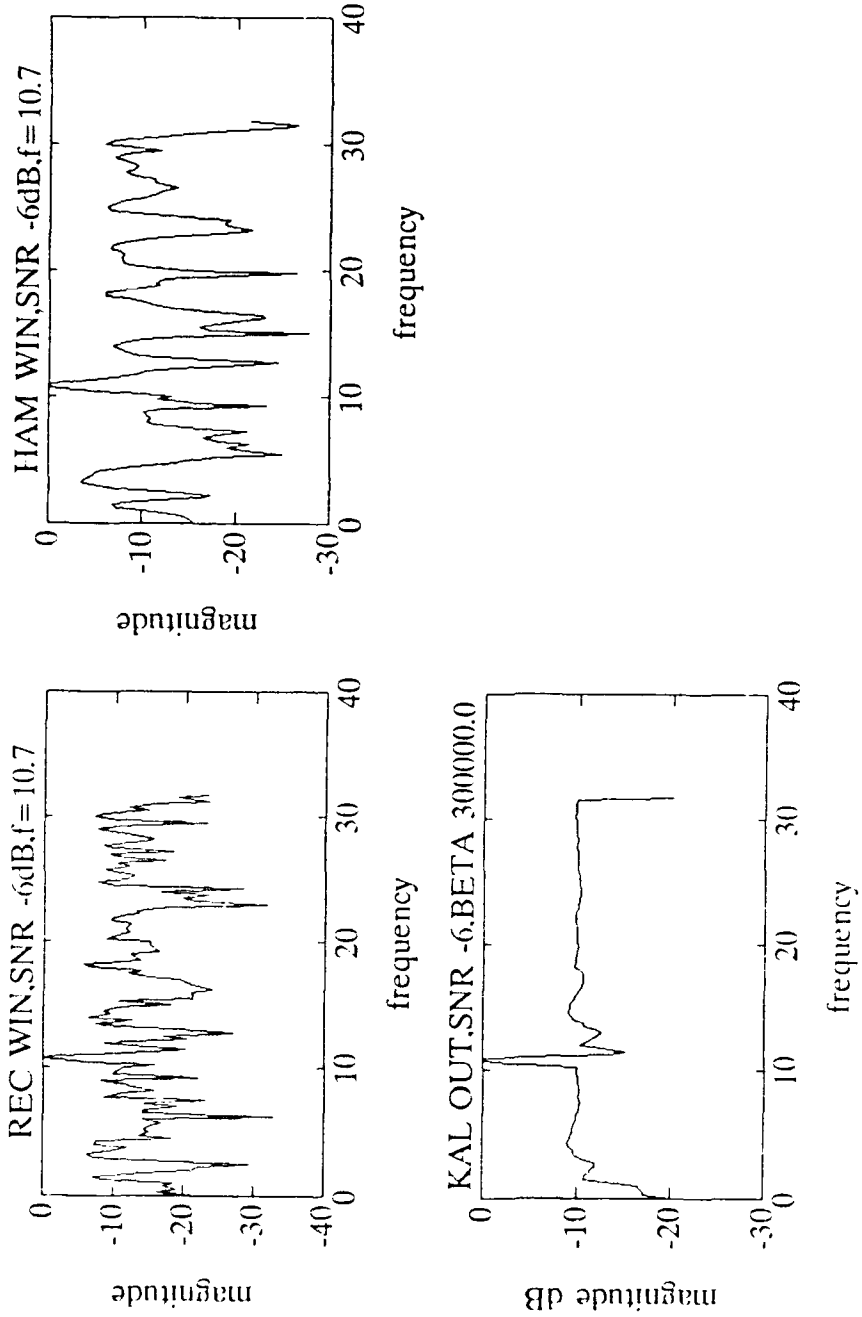


Figure C.14. -6 dB Input SNR, Noise Realization 4

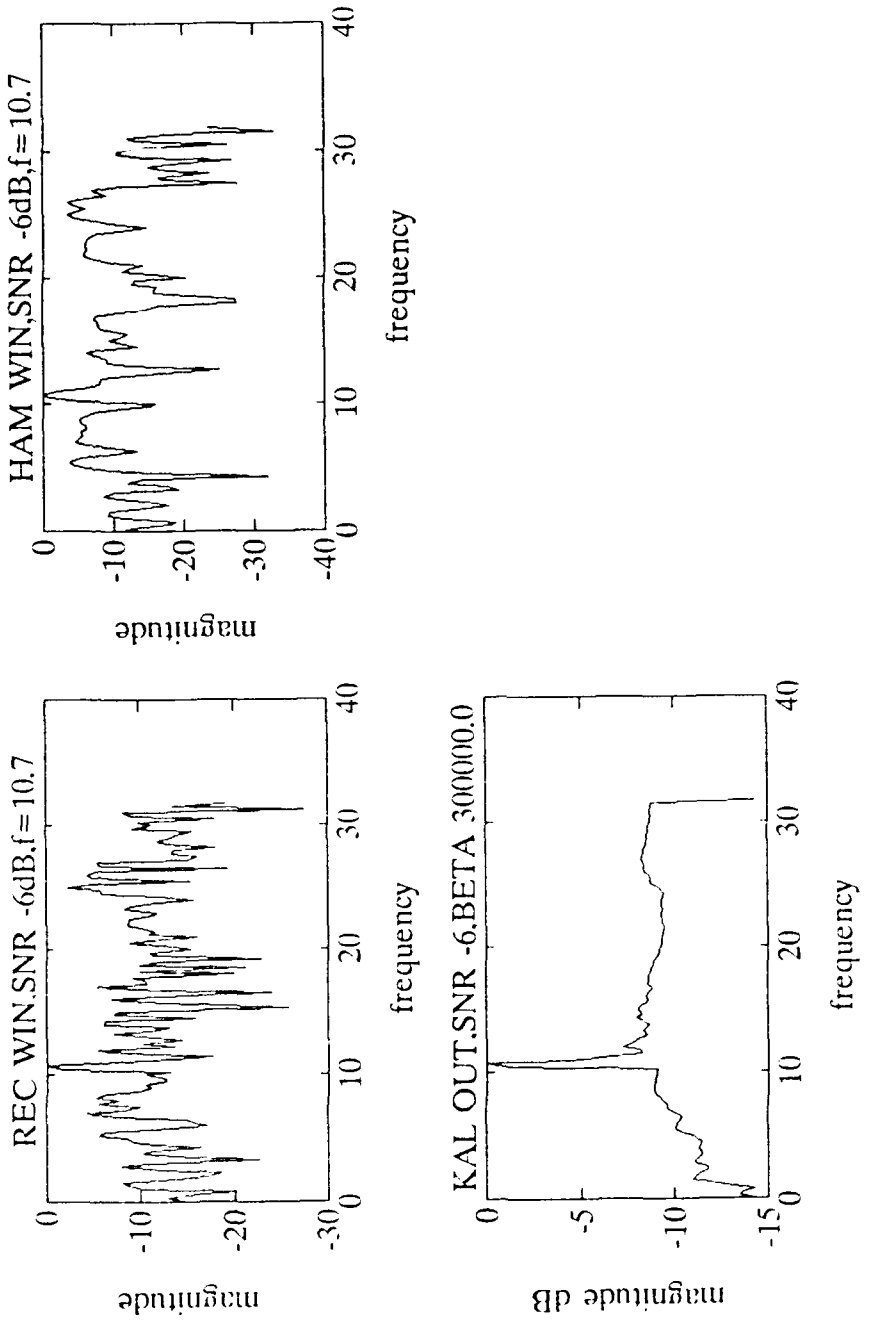


Figure C.15. -6 dB Input SNR, Noise Realization 5

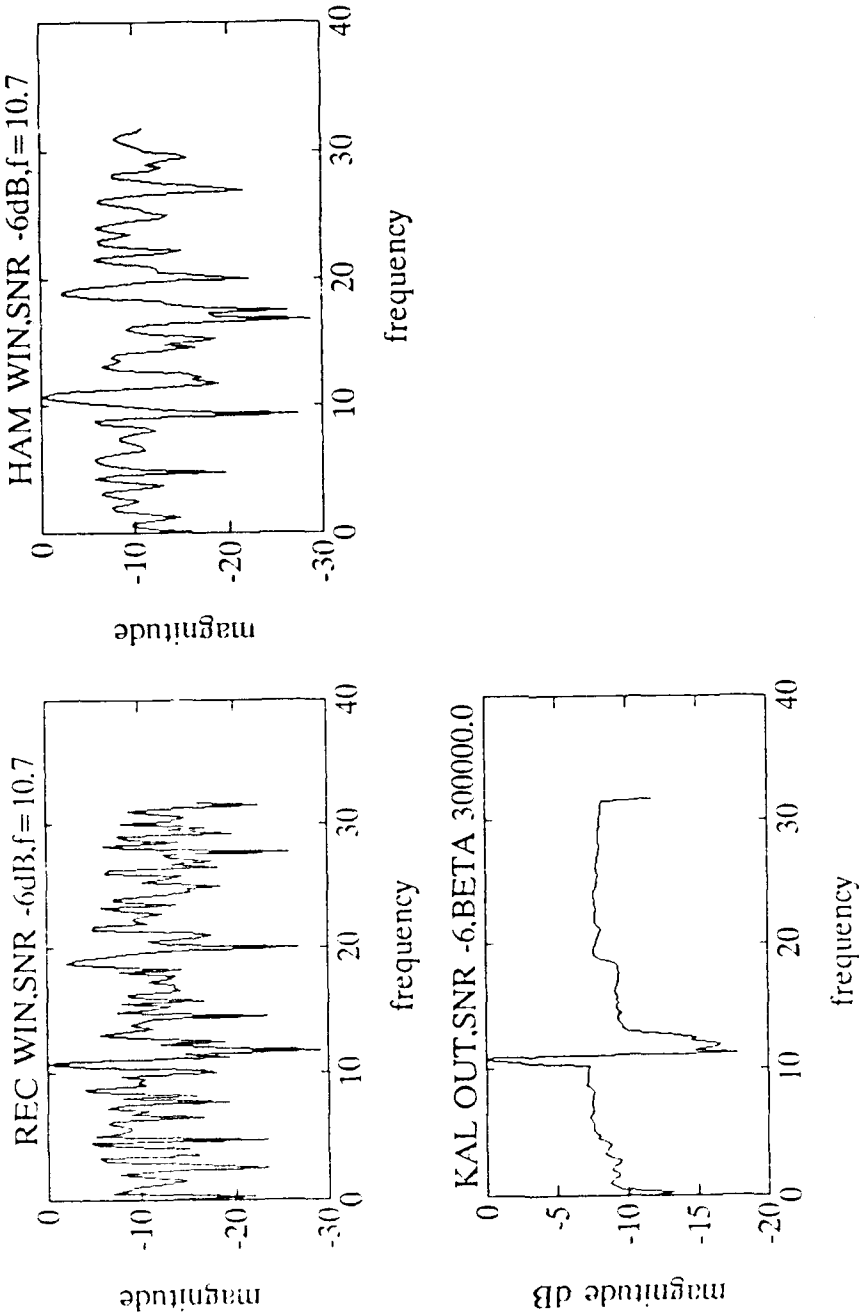


Figure C.16. -6 dB Input SNR, Noise Realization 6

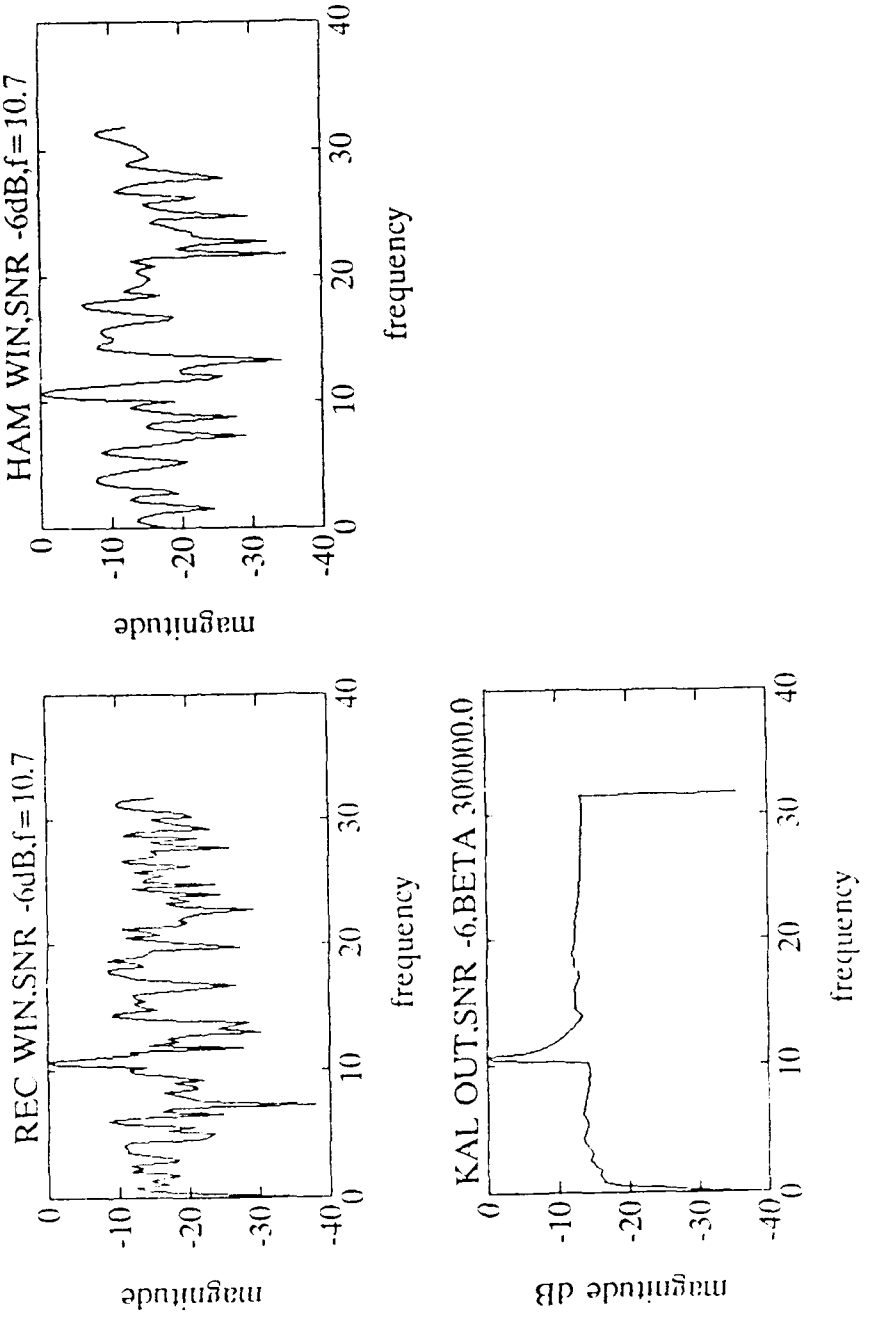


Figure C.17. -6 dB Input SNR, Noise Realization 7

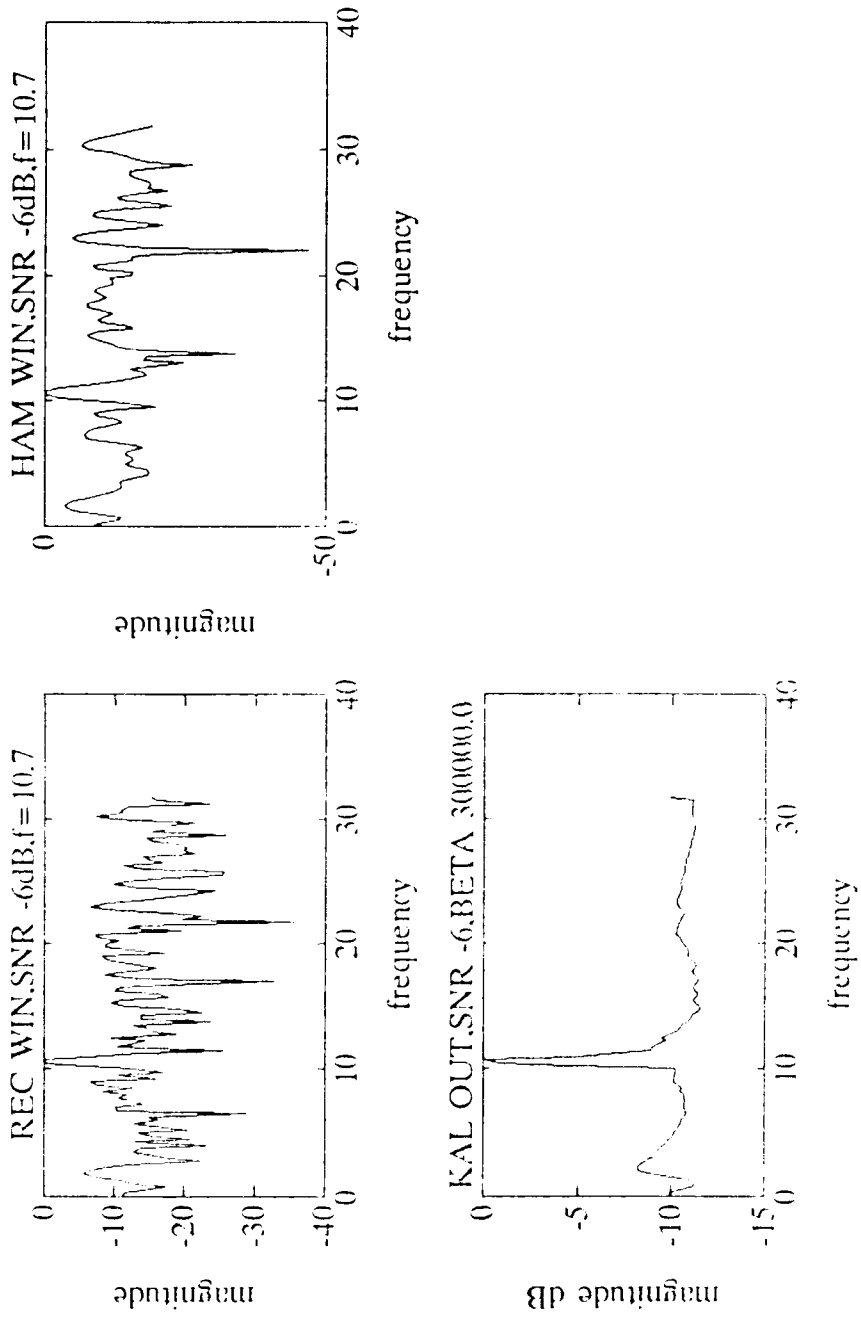


Figure C.18. -6 dB Input SNR, Noise Realization 8

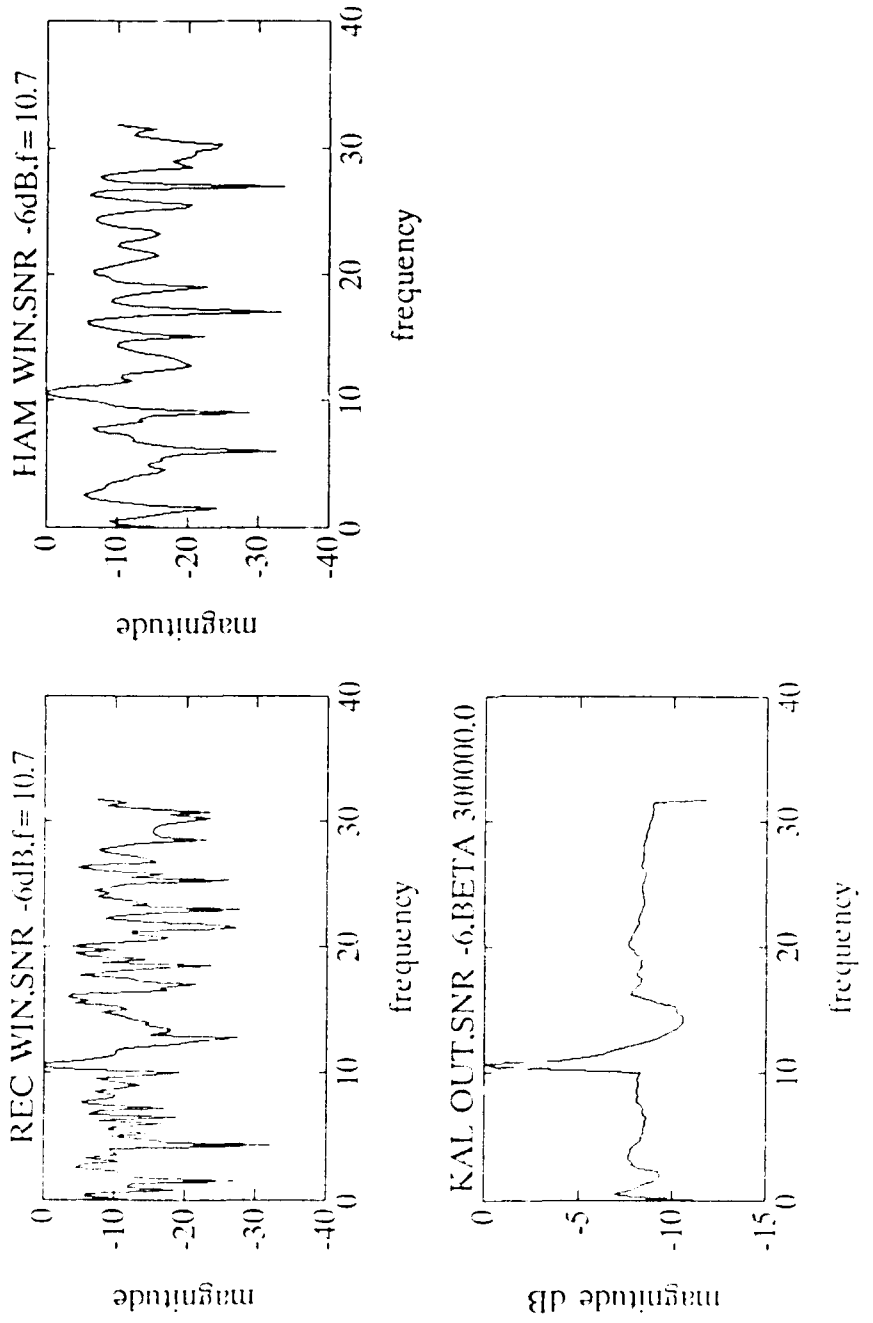


Figure C.19. -6 dB Input SNR, Noise Realization 9

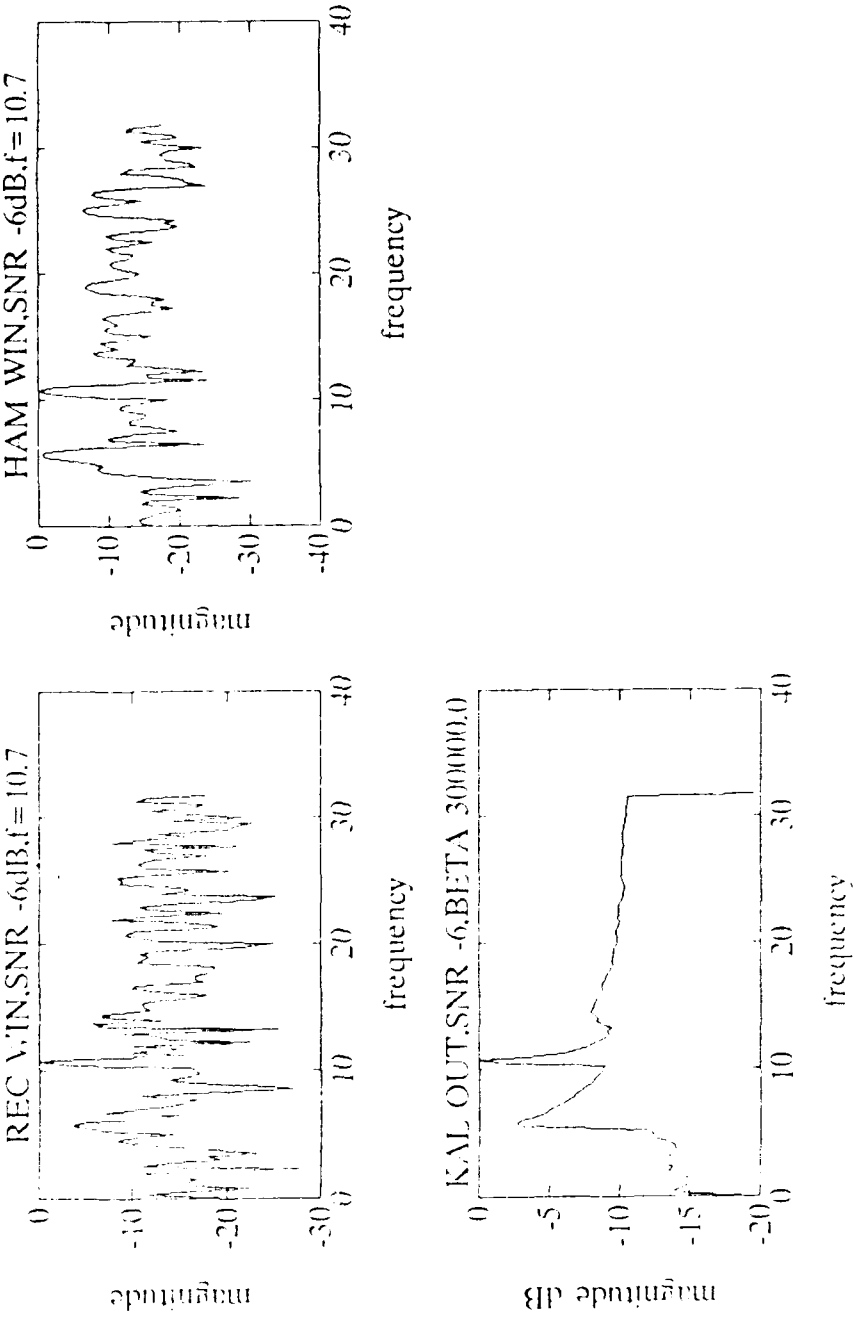


Figure C.20. -9 dB Input SNR, Noise Realization 10

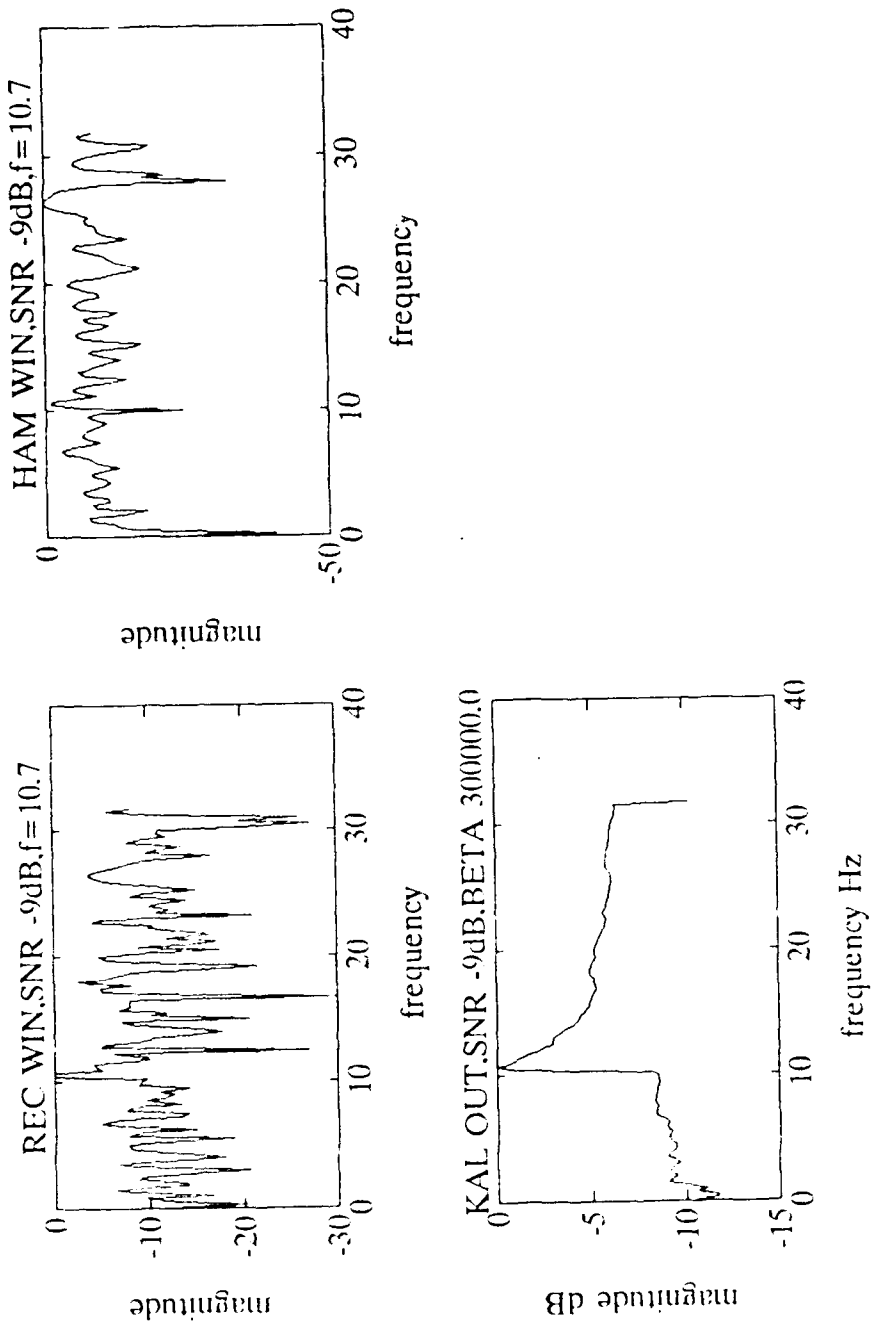


Figure C.21. -9 dB Input SNR, Noise Realization 1

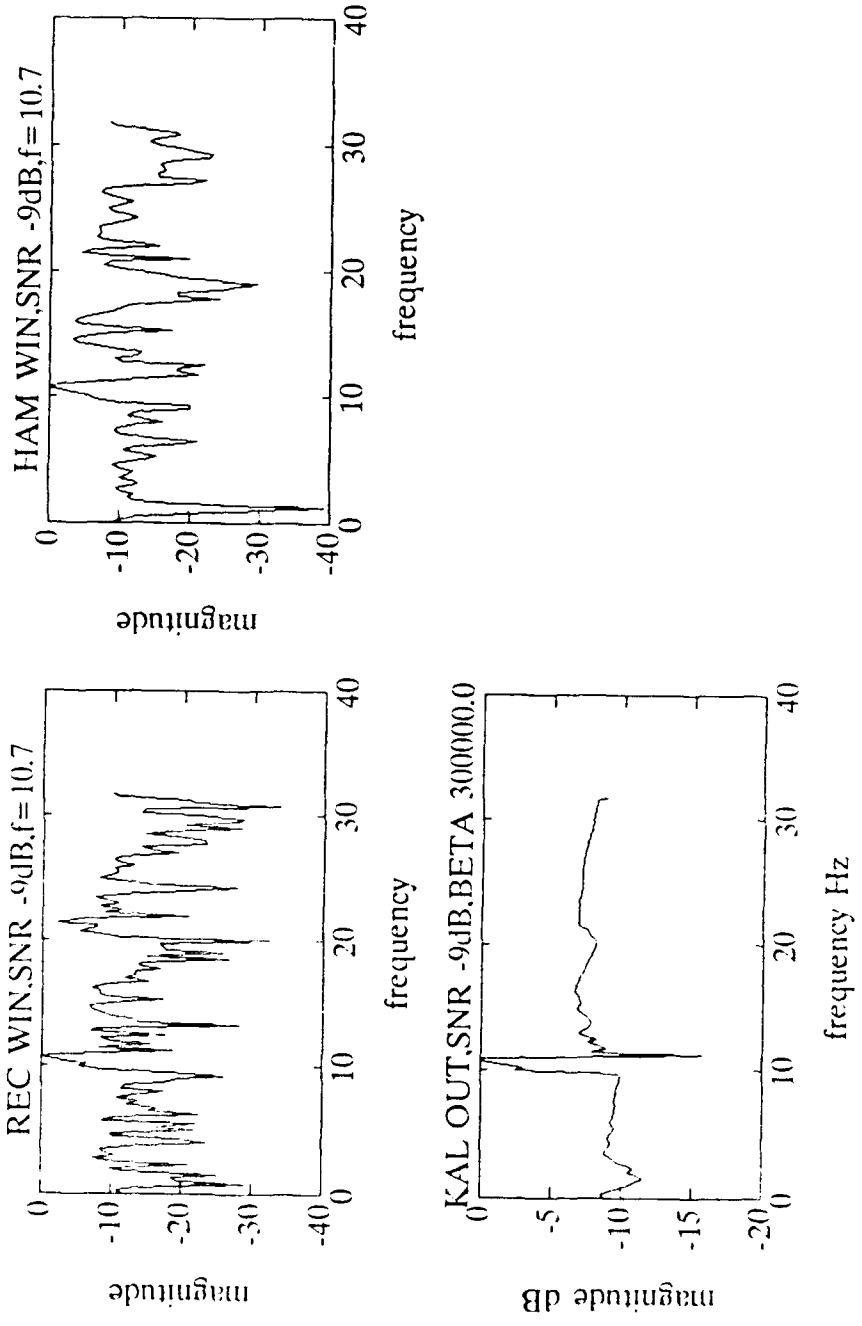


Figure C.22. -9 dB Input SNR, Noise Realization 2

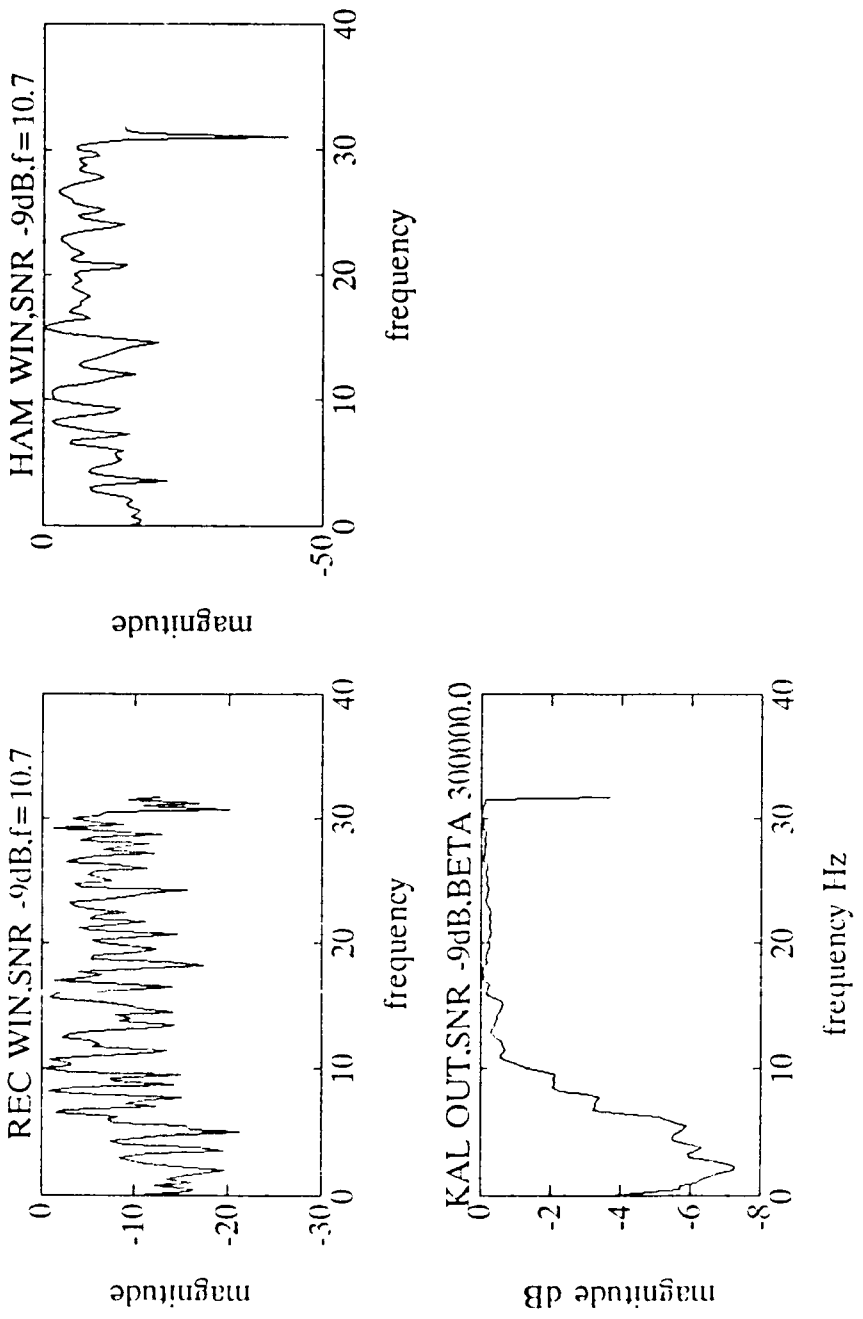


Figure C.23. -9 dB Input SNR, Noise Realization 3

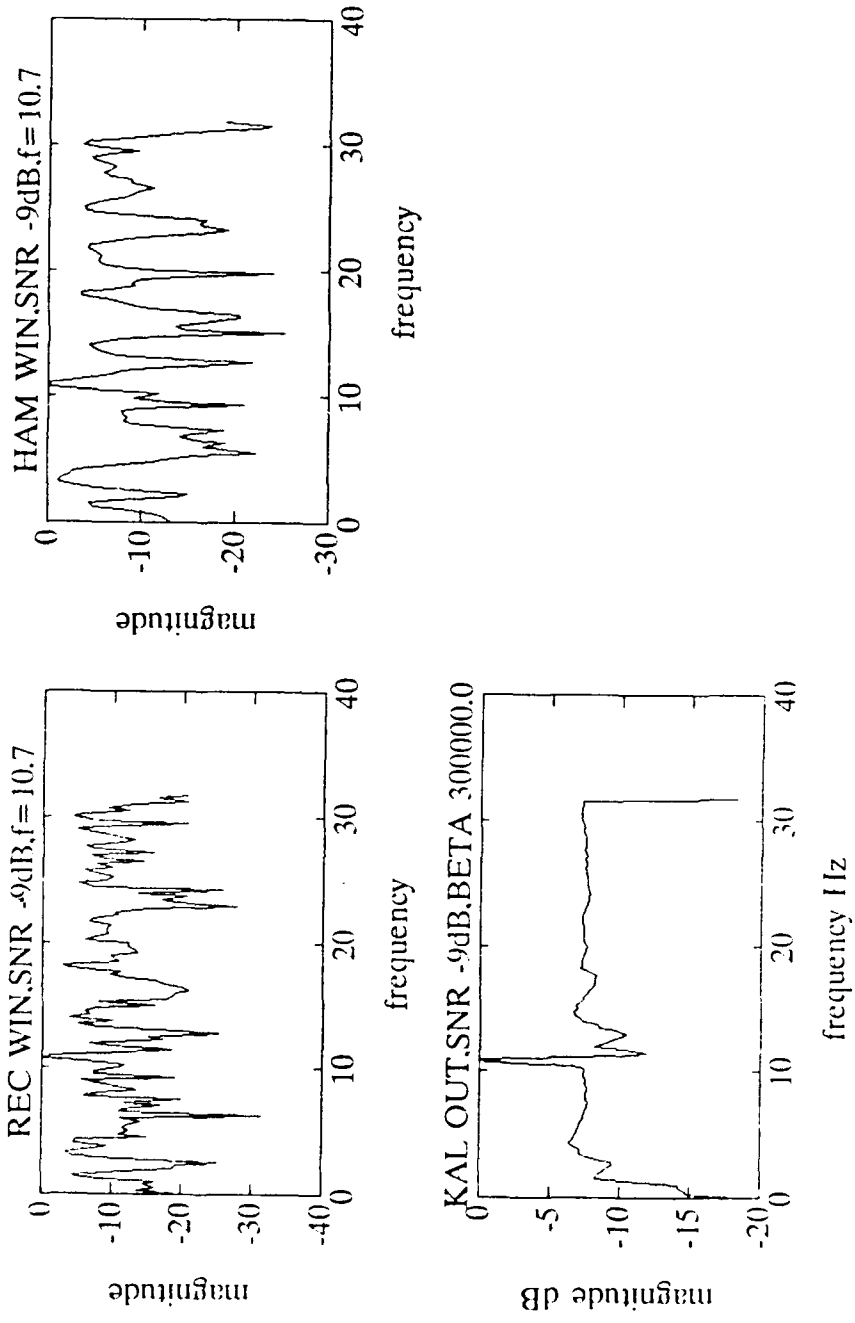


Figure C.24. -9 dB Input SNR, Noise Realization 4

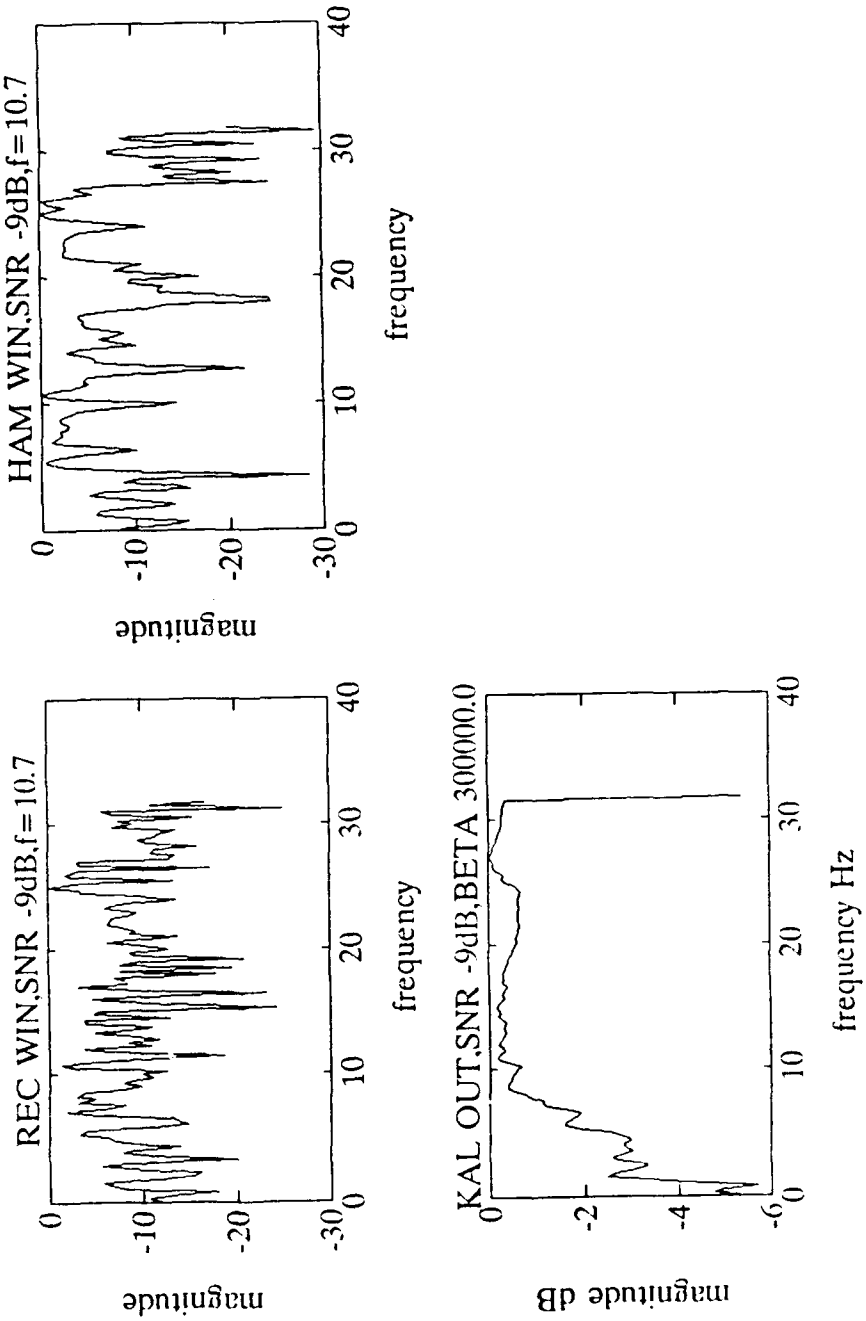


Figure C.25. -9 dB Input SNR, Noise Realization 5

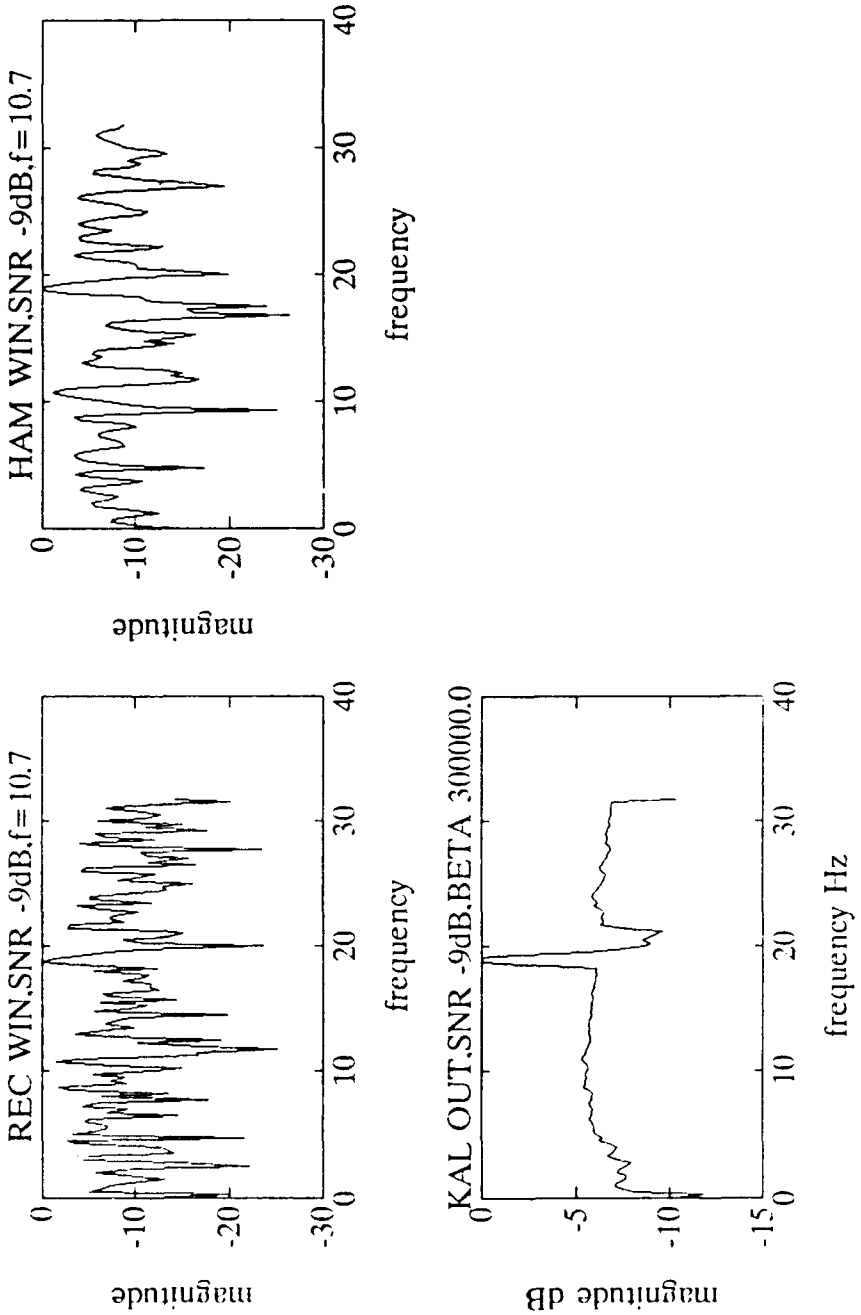


Figure C.26. -9 dB Input SNR, Noise Realization 6

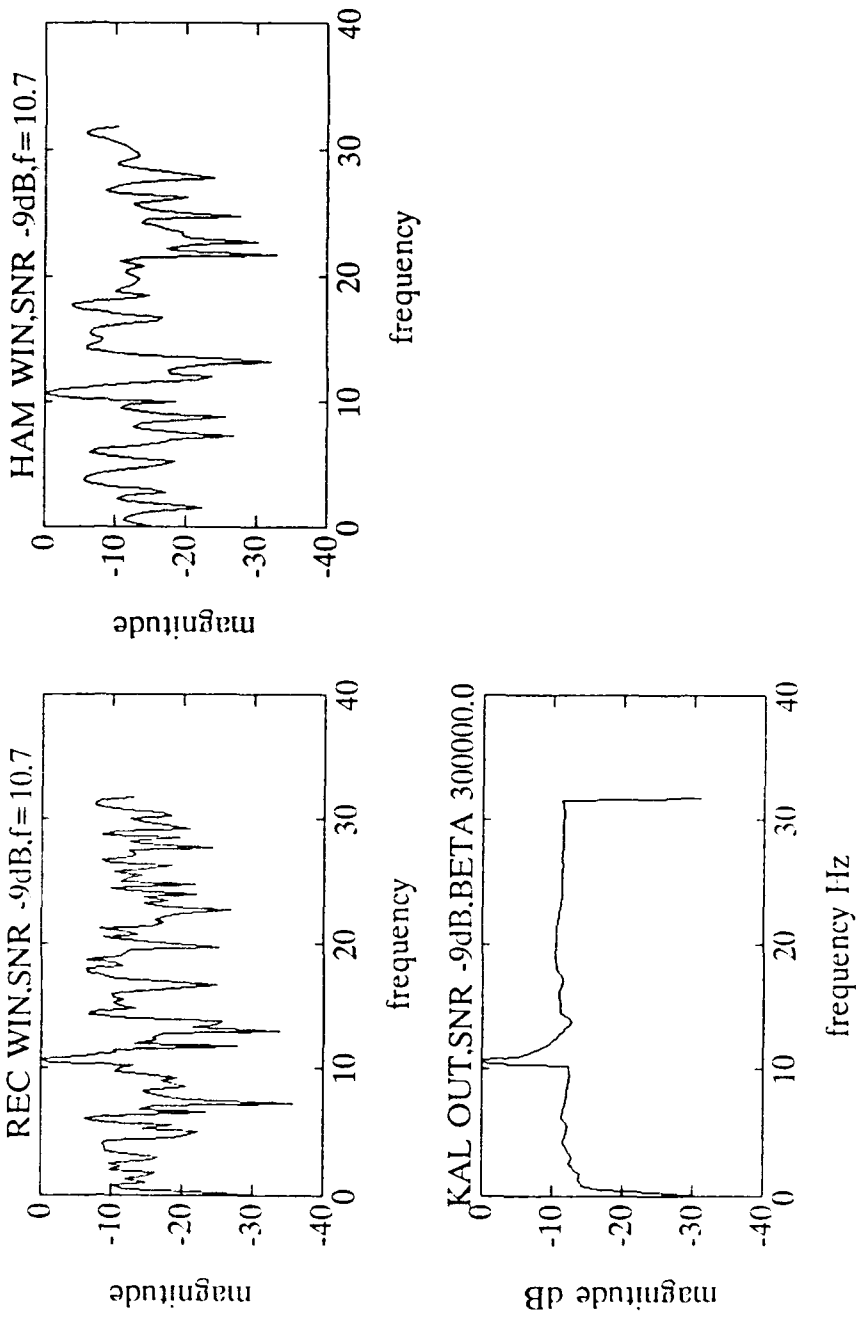


Figure C.27. -9 dB Input SNR, Noise Realization 7

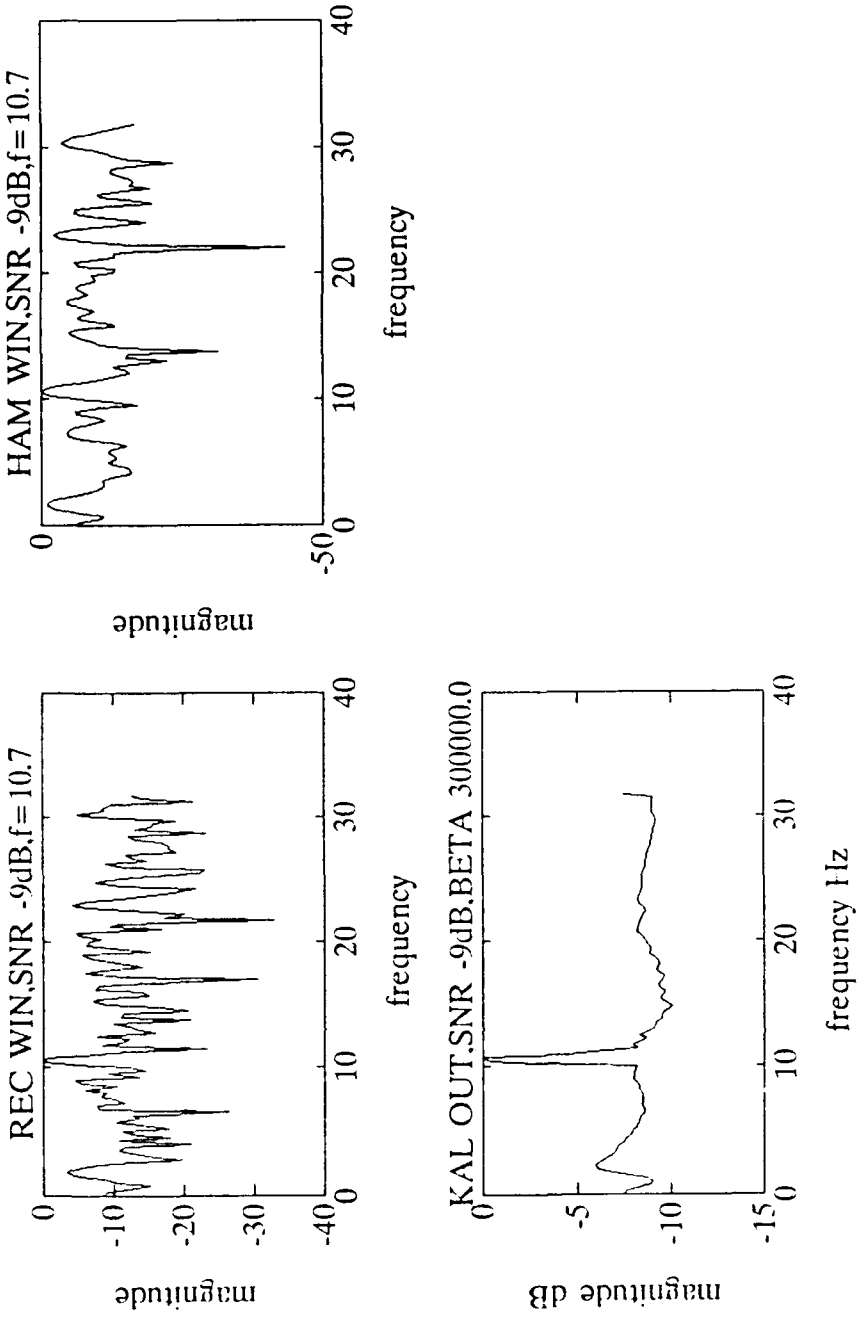


Figure C.28. -9 dB Input SNR, Noise Realization 8

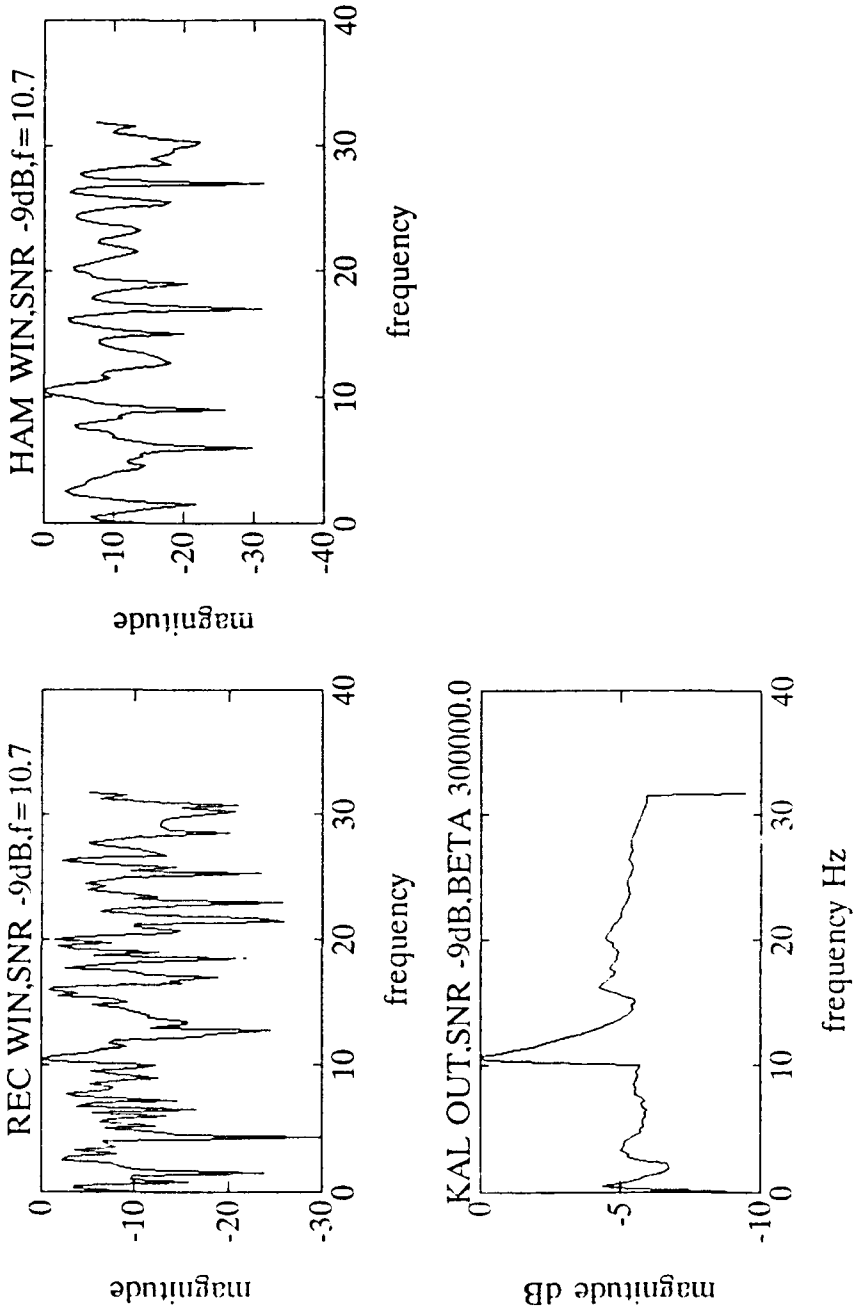


Figure C.29. -9 dB Input SNR, Noise Realization 9

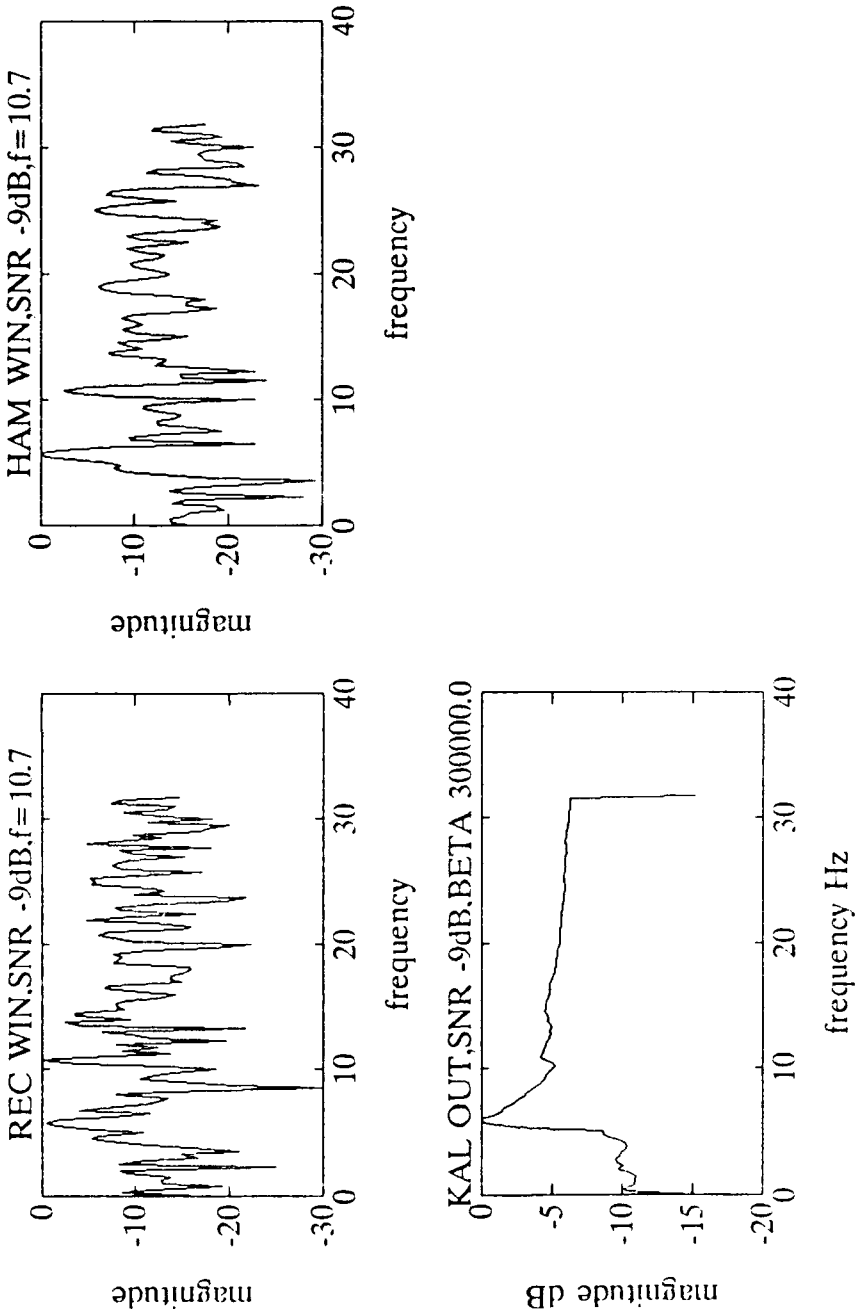


Figure C.30. -9 dB Input SNR, Noise Realization 10

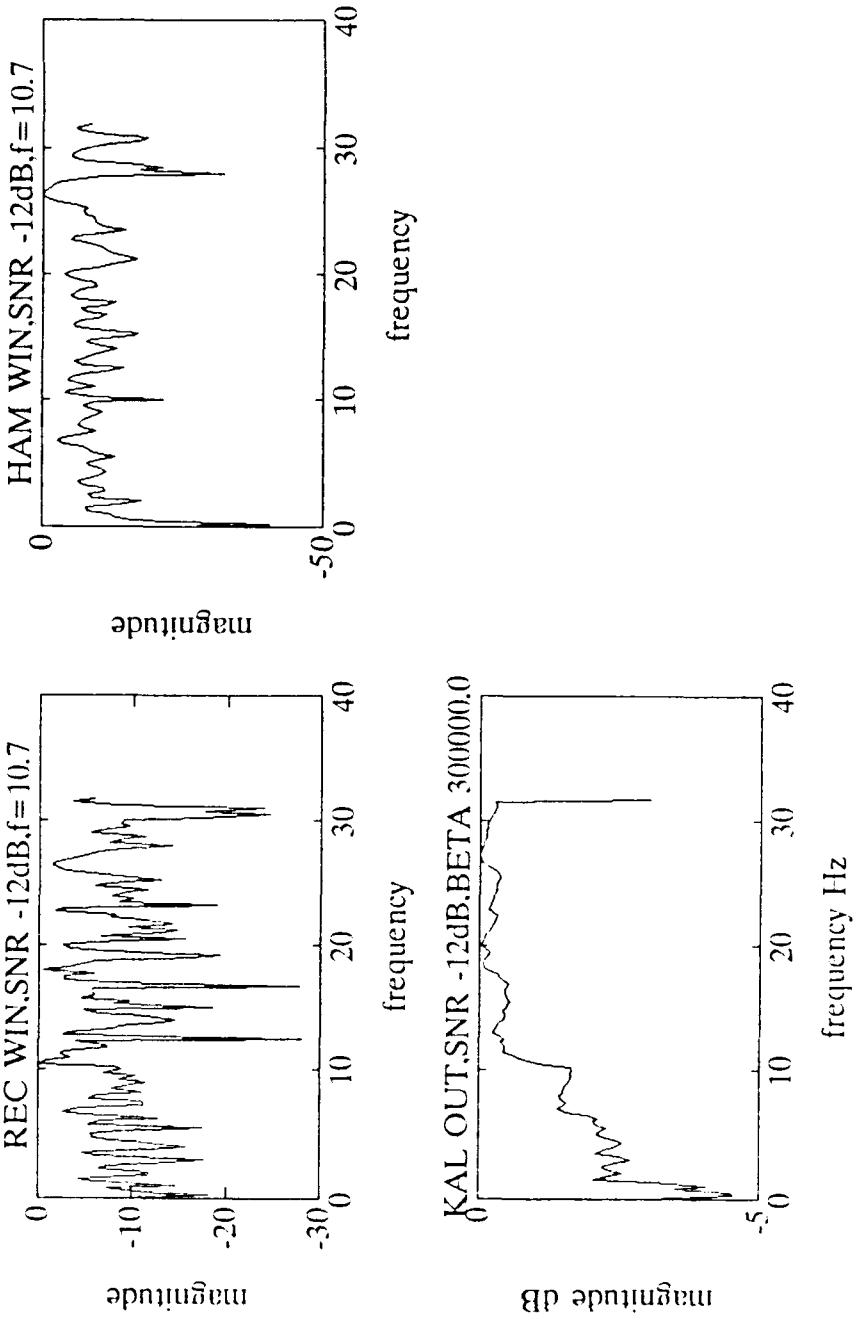


Figure C.31. -12 dB Input SNR, Noise Realization 1

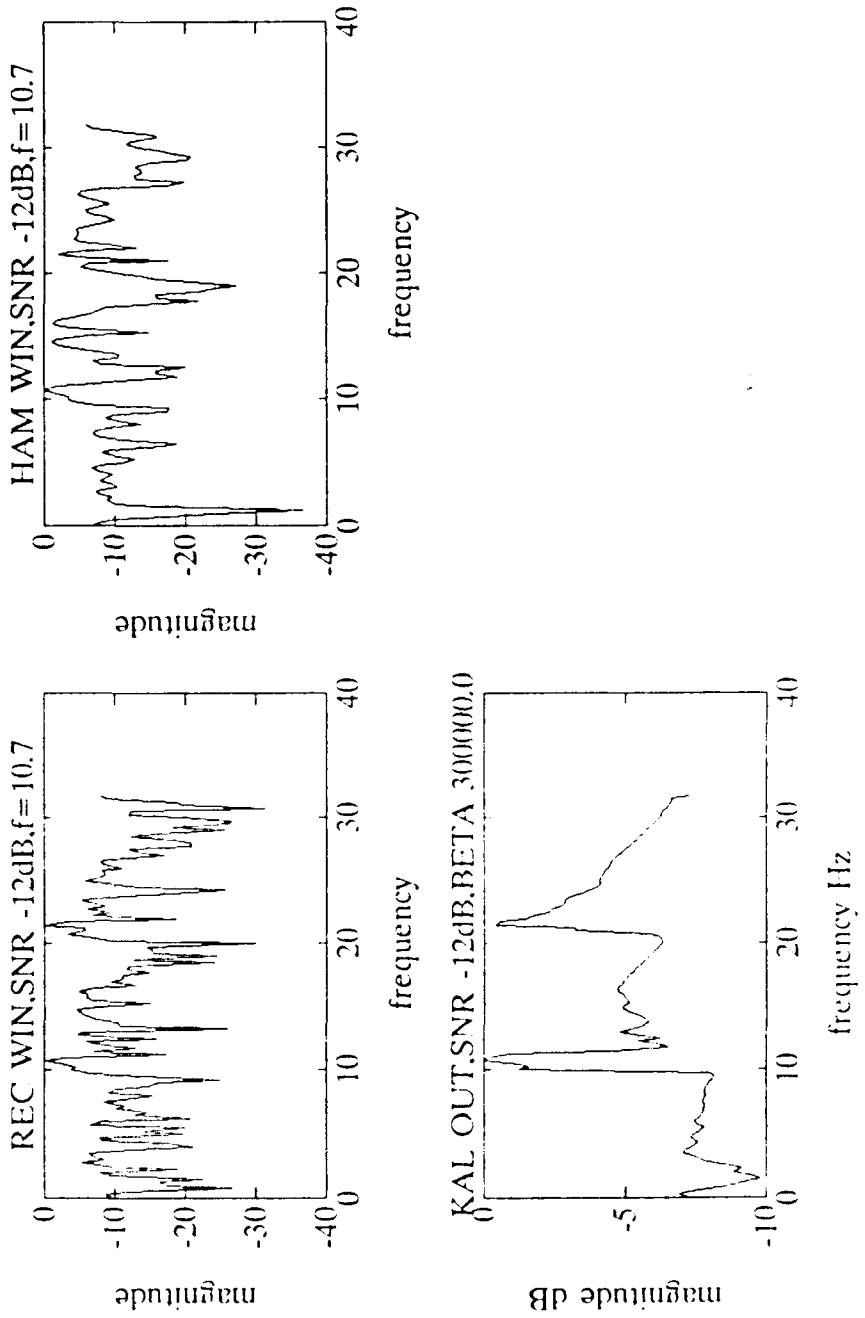


Figure C.32. -12 dB Input SNR, Noise Realization 2

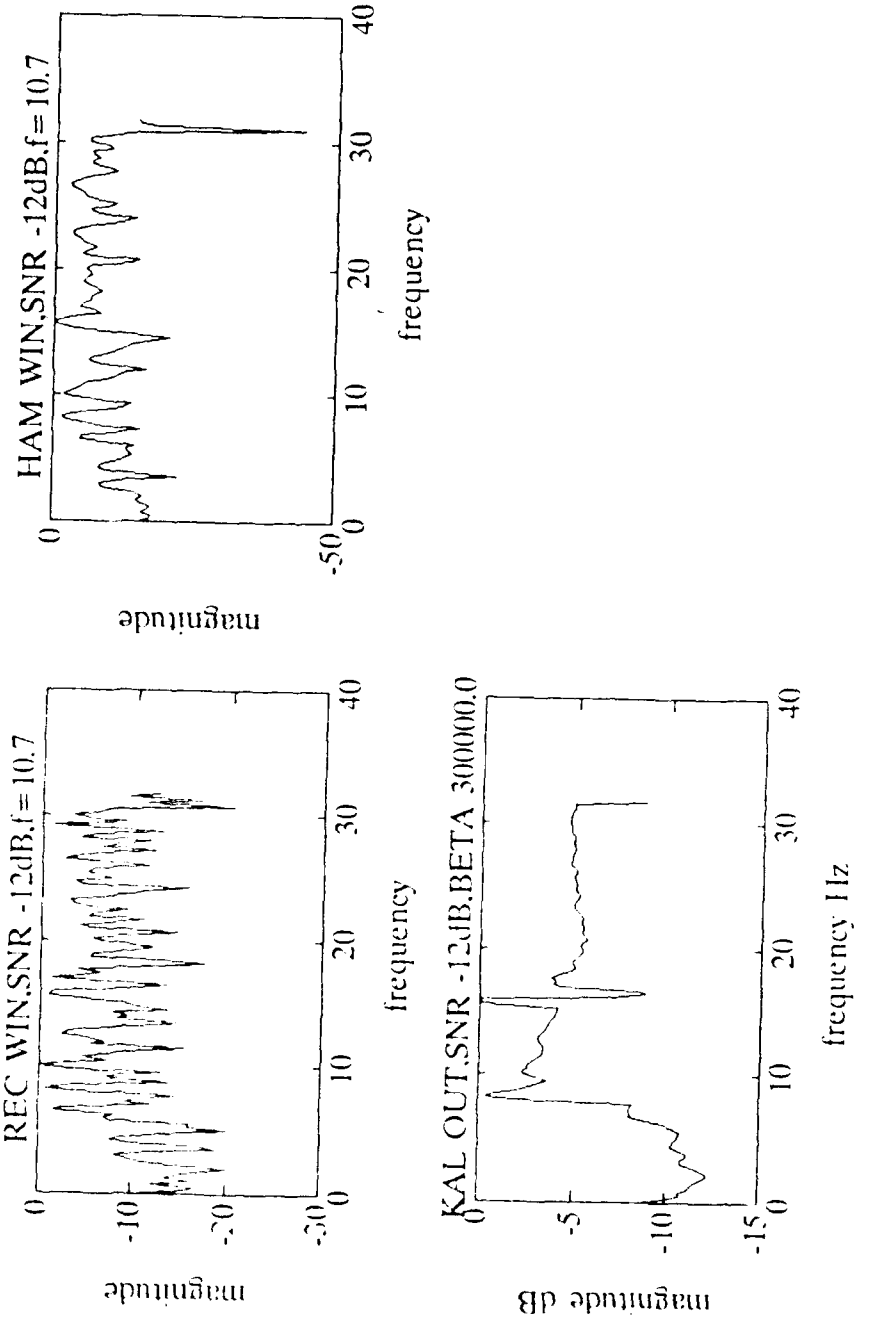


Figure C.33. -12 dB Input SNR, Noise Realization 2

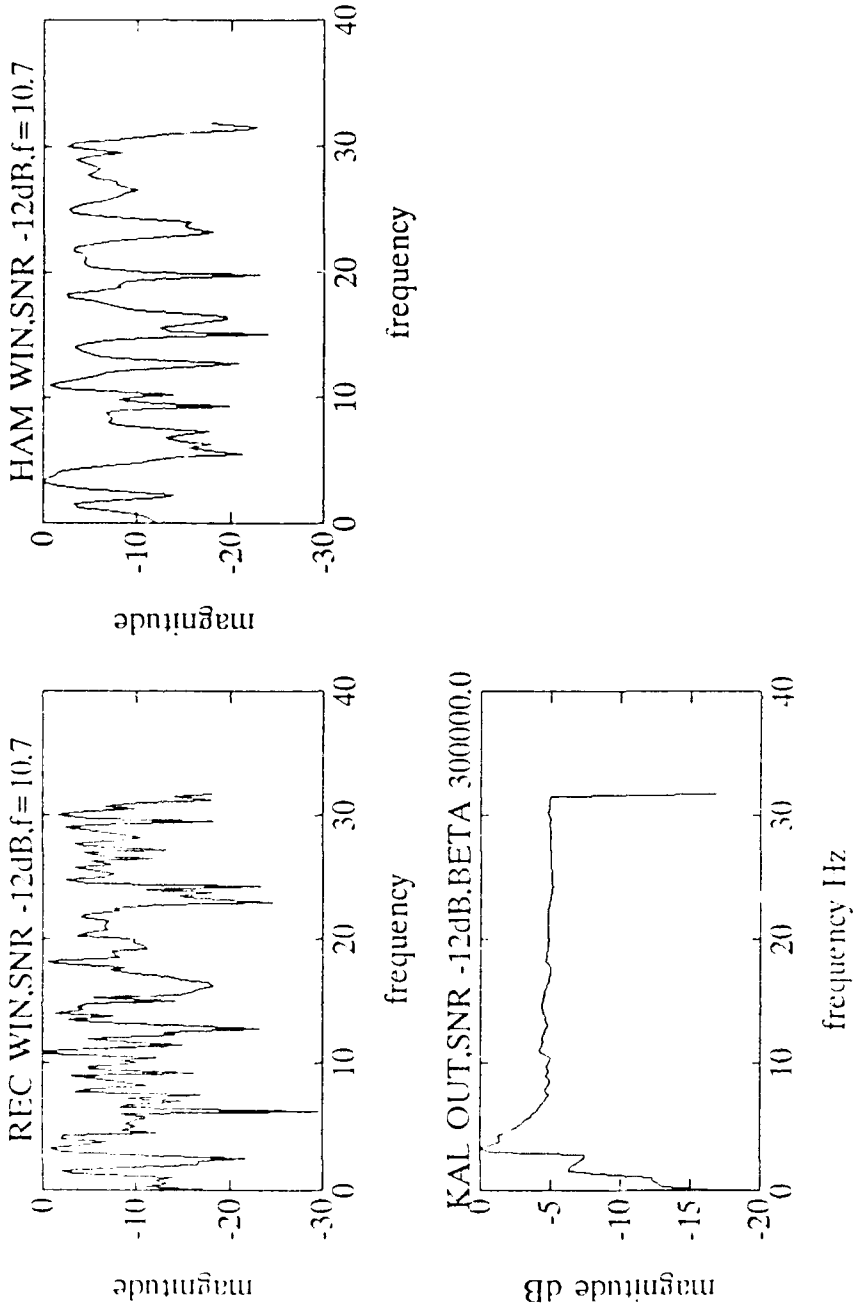


Figure C.34. -12 dB Input SNR, Noise Realization 4

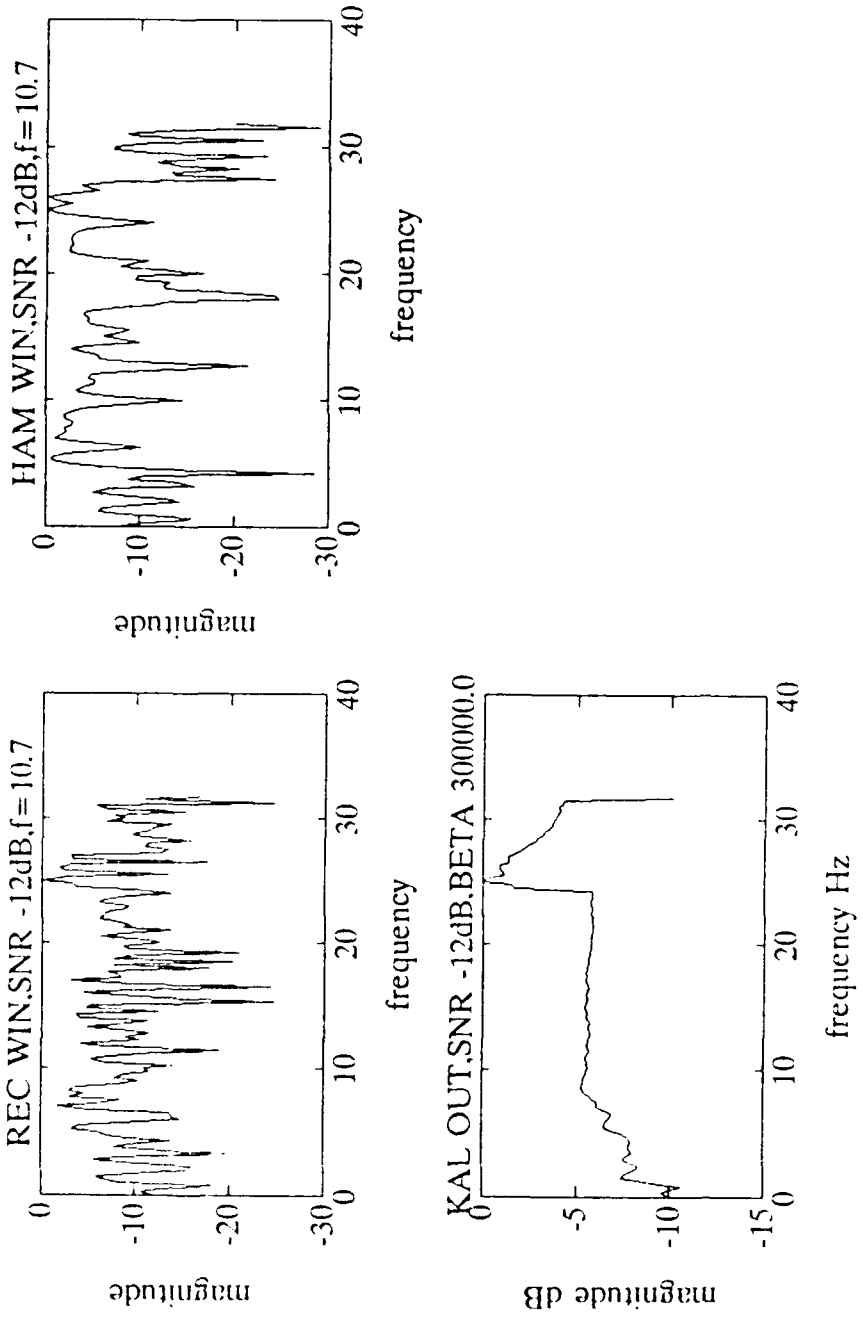


Figure C.35. -12 dB Input SNR, Noise Realization 5

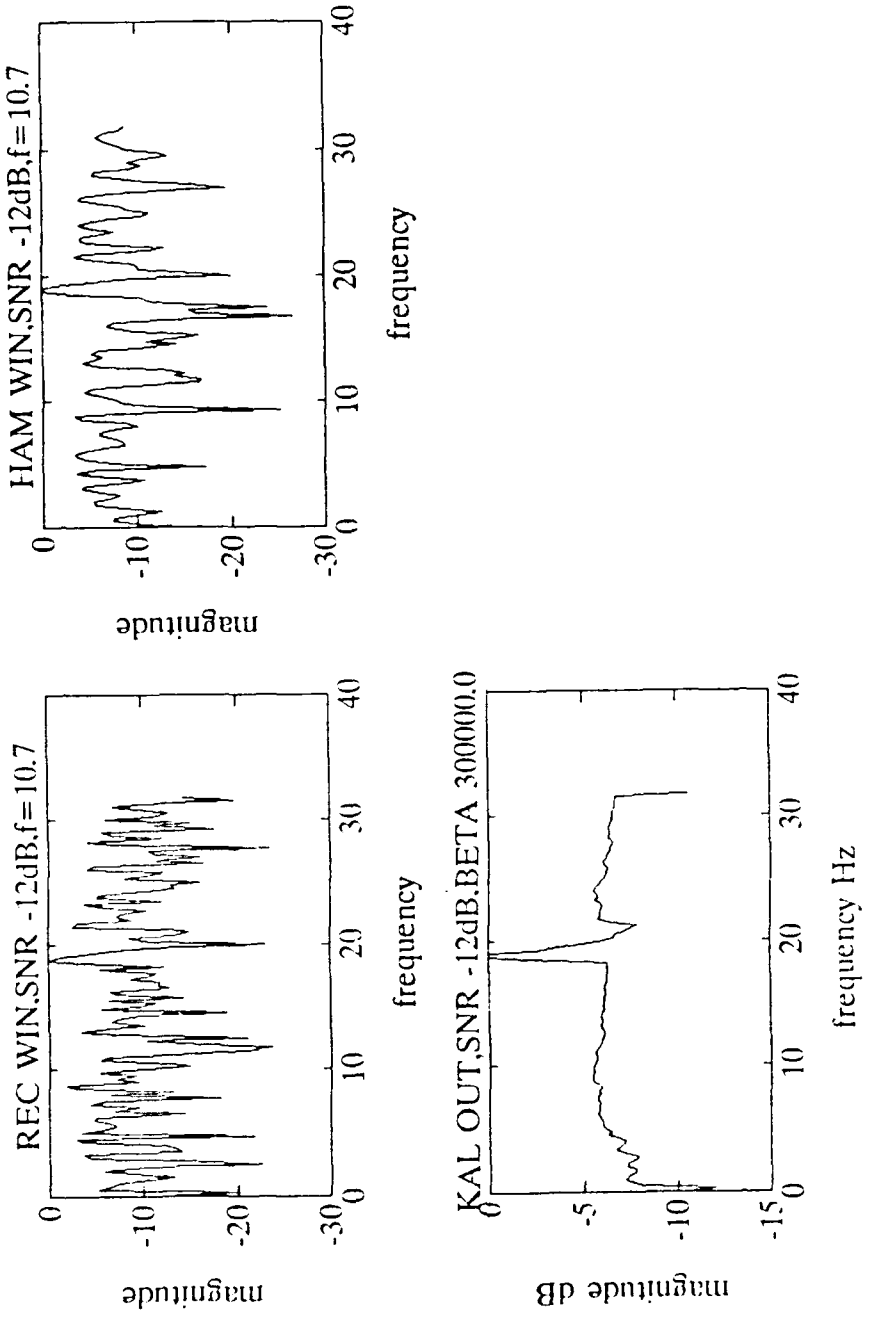


Figure C.36. -12 dB Input SNR, Noise Realization 6

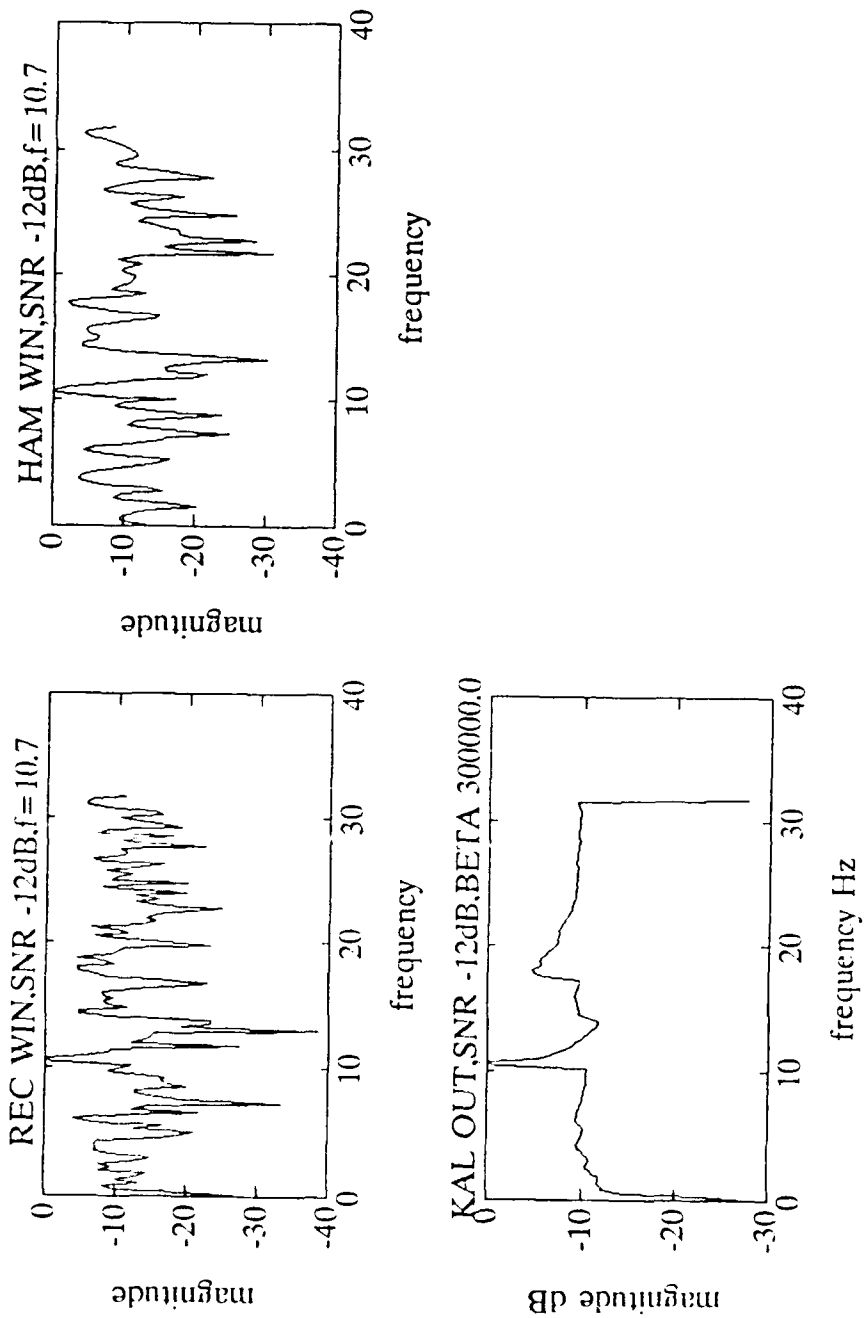


Figure C.37. -12 dB Input SNR, Noise Realization 7

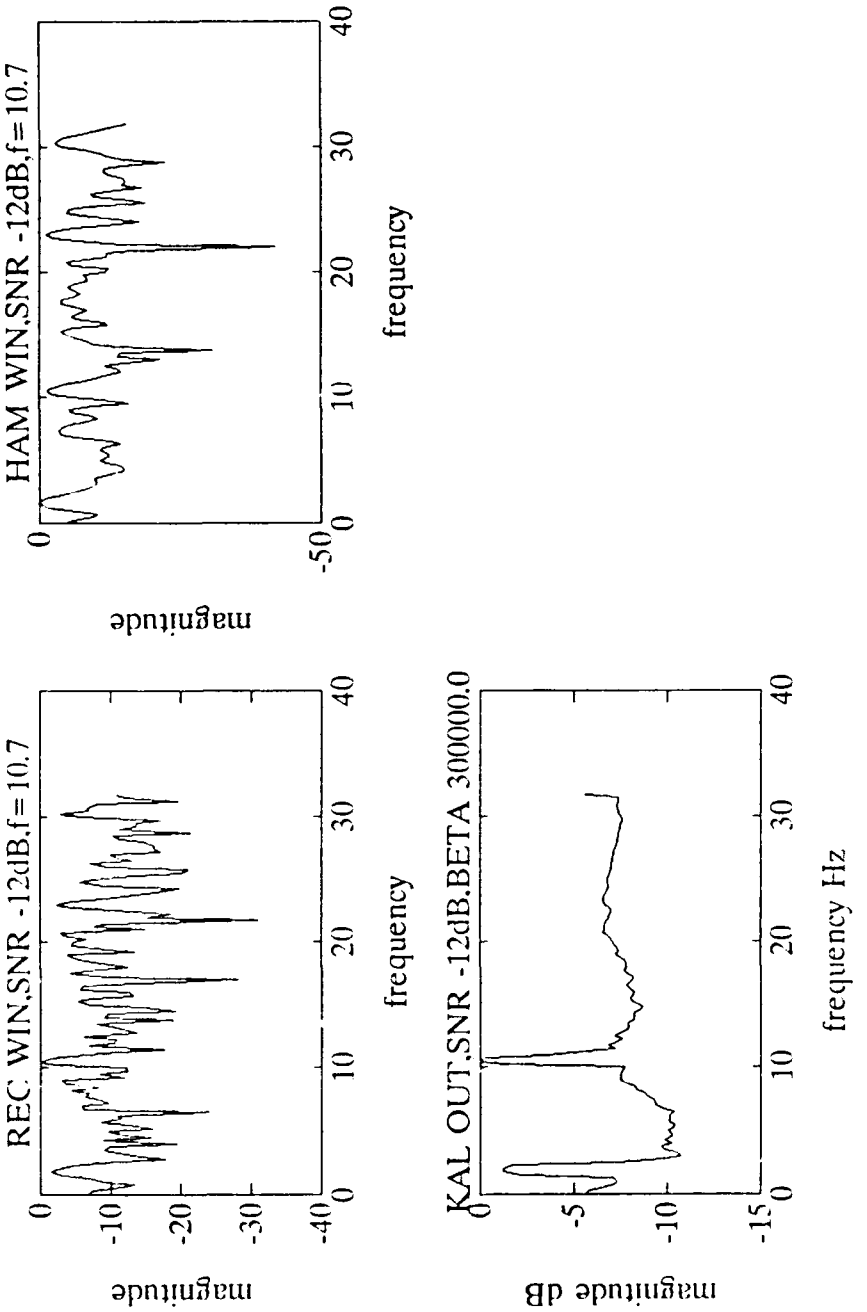


Figure C.38. -12 dB Input SNR, Noise Realization 8

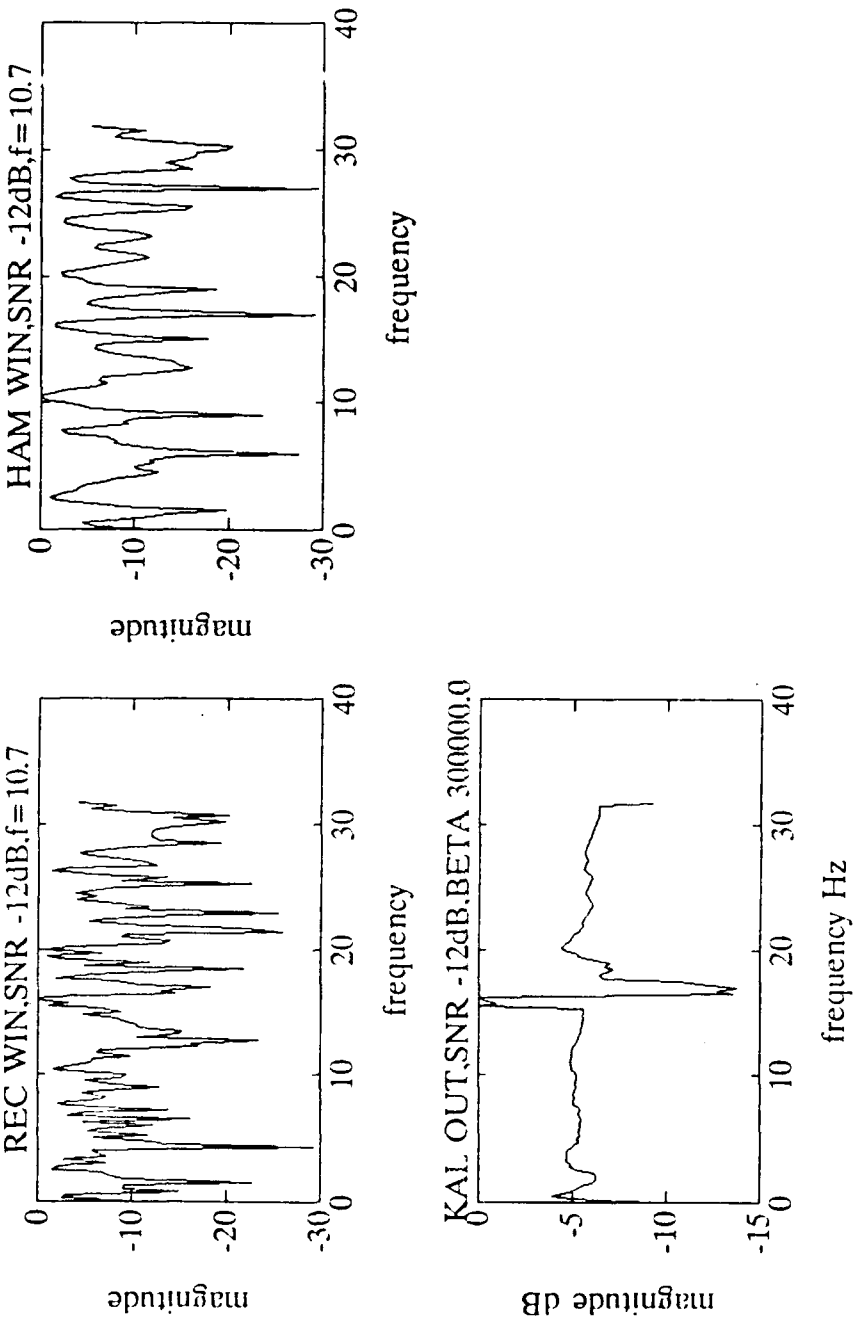


Figure C.39. -12 dB Input SNR, Noise Realization 9

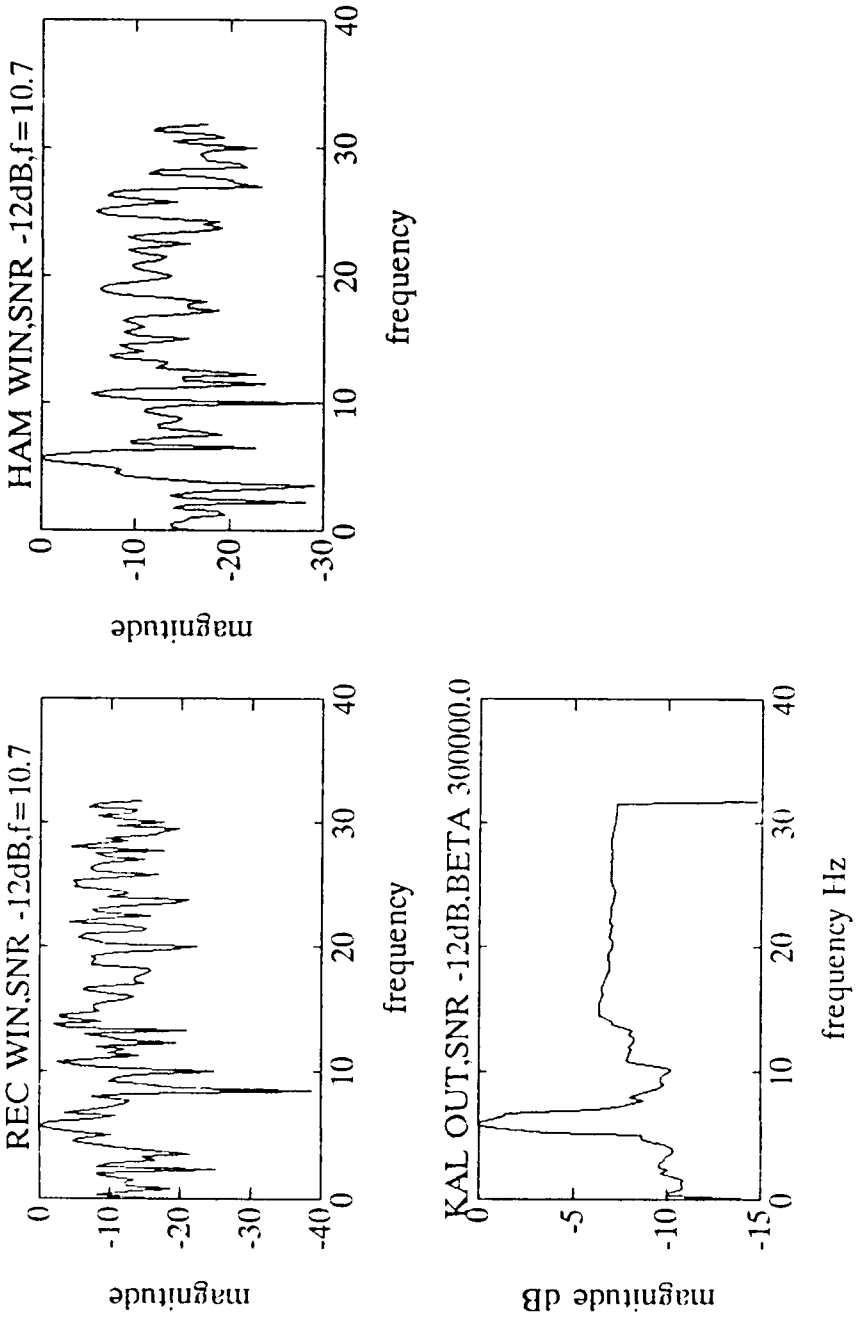


Figure C.40. -12 dB Input SNR, Noise Realization 10

APPENDIX D
EFFECTS OF DATA/TRANSFORM LENGTH

The effects of varying the data/transform length on the performance of the Kalman filter were investigated. The baseline test case was a pair of sinusoids, frequencies 10.7 and 13.9 Hz (not at FFT bin centers) embedded in Gaussian white noise. Data records of 128, 512, and 1024 points were zero-padded to twice their original length. As discussed in Chapter III, the time series input SNR was decreased with increasing data/transform length in order to compensate for the higher processing gains of the longer data records (so as to maintain the SNR at the input to the Kalman filter at approximately 12 dB). A β of 300,000 was used since this value gave good results with the baseline 128 data point test case. Results are given in Figures D.1 through D.3. For comparison, the unfiltered periodograms are also shown for each data/transform length. Results indicate that as data transform length is increased, β may have to be increased in order to obtain optimum smoothing of the spectral estimate. The dependence of β upon transform/data length is a potential topic for follow-on study.

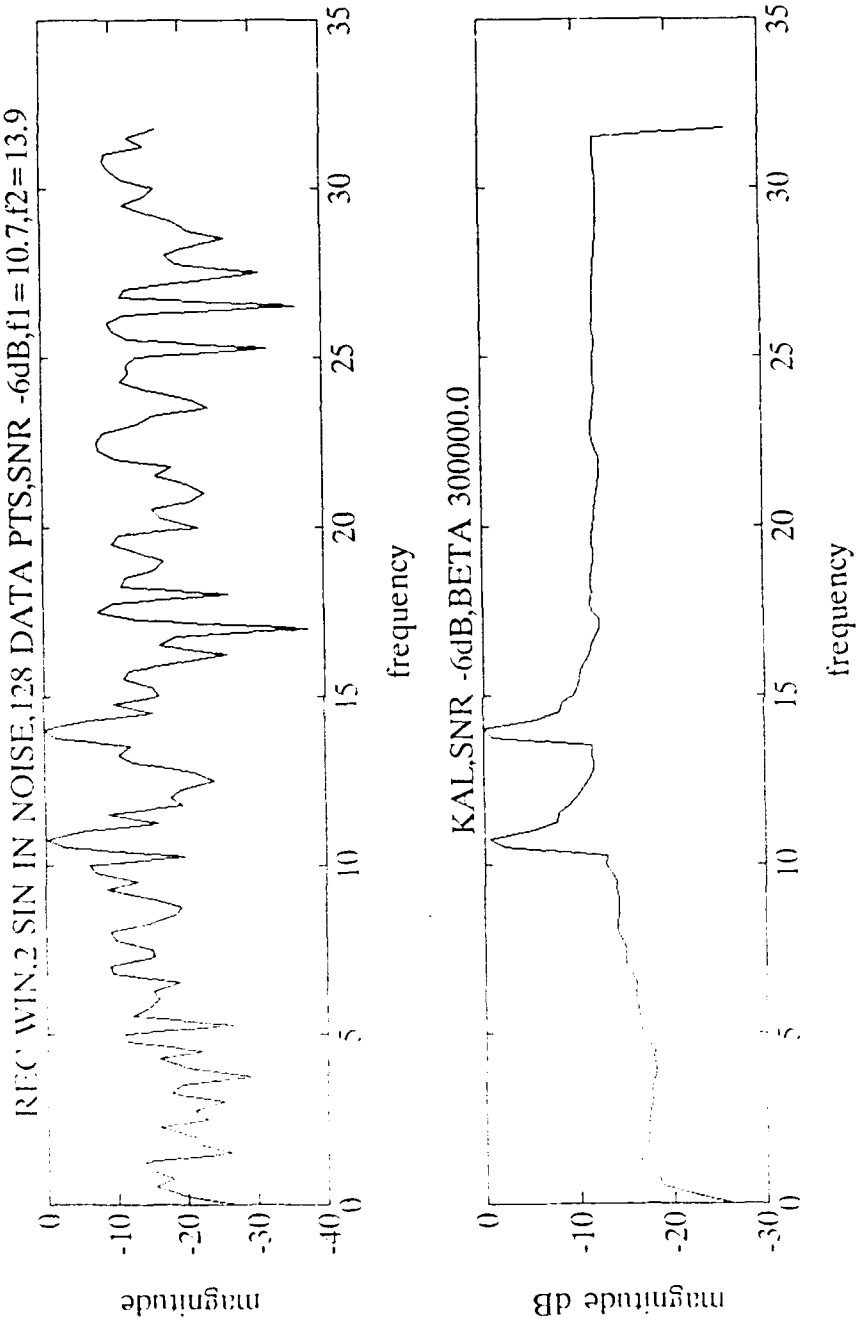


Figure D.1. 128 Data Points

Figure D.2. 512 Data Points

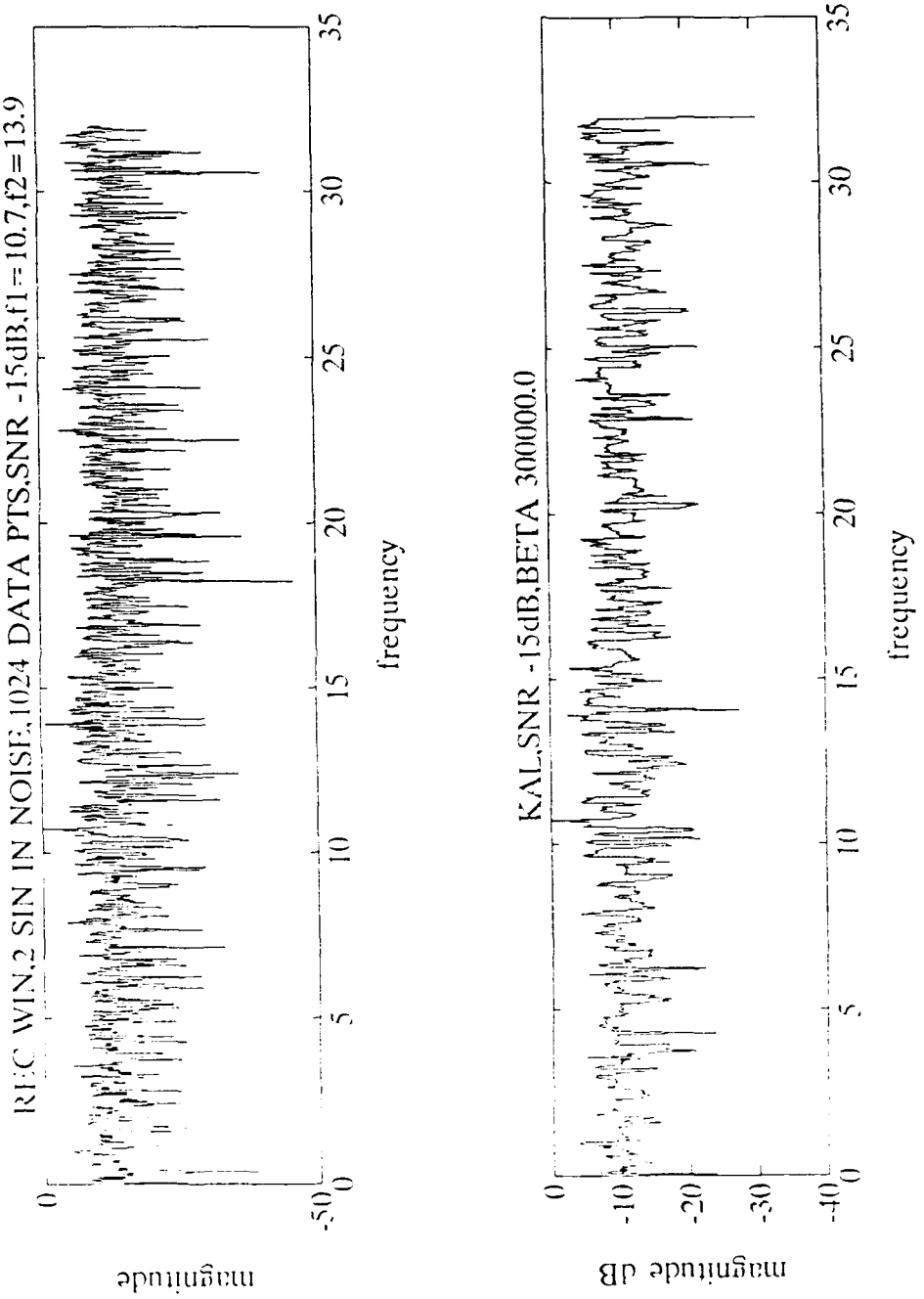


Figure D.3. 1024 Data Points

REFERENCES

1. Harris F.J., On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proceedings of the IEEE, Vol. 66, No. 1, pp. 51-83, January 1978.

2. Blackman, R.B. and Tukey, J.W., The Measurement of Power Spectra, Appendix B.5, pp. 95-100, New York:Dover, 1958.

3. Kay, M.S. and Marple, S.L., Spectrum Analysis - A Modern Perspective, Proceedings of the IEEE, Vol. 69, No. 11, pp. 1380-1414, November 1981.

4. Priestly, M.B., Spectral Analysis and Time Series, Vol. 1, pp. 66-67 and 397-405, Academic Press, London, 1981.

5. Welch, P.D., The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms, IEEE Transactions, Audio Electroacoust., Vol. AU-15, pp. 70-73, June 1967.

6. Jenkins, G.M. and Watts, D.G., Spectral Analysis and Its Applications, Ch. 6, pp. 209-257, Hoden-Day, San Francisco, 1968.

7. Marple, S.L. Jr., Digital Spectral Analysis with Applications, p. 153, Prentice-Hall, New Jersey, 1987.

8. Kalman, R.E., A New Approach to Linear Filtering and Prediction Problems, Trans. ASME (J. Basic Engineering), Vol. 82D, No. 1, pp. 35-45, March 1960.

9. Kalman, R.E. and Bucy, R.S., New Results in Linear Filtering and Prediction Theory, Trans. ASME (J. Basic Engineering), Vol. 83D, No. 1, pp. 95-108, March 1961.

10. Schwartz, M. and Shaw, L., Signal Processing, Discrete Spectral Analysis, Detection and Estimation, Chs. 6 and 7, pp. 274-385, McGraw-Hill, New York, 1975.

DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Curricular Officer, Code 32 Electronics and Communications Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Prof. Ralph Hippenstiel (Code EC/Hi) Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	3
6.	Prof. Roberto Cristi (Code EC/Cx) Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
7.	Prof. Jeff Burl (Code EC/Bl) Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
8.	LT J.M. Jorgensen, USN SMC 1872 Naval Postgraduate School Monterey, CA 93943-5000	1
9.	CAPT William W. Go, USMC Department of Physics U.S. Naval Academy Annapolis, MD 21402	1
10.	Naval Ocean Systems Center ATTN: Dr. C.E. Persons, Code 732 San Diego, CA 92152	1

11. Commandant of the Marine Corps 1
Code TE 06
Headquarters, U.S. Marine Corps
Washington, DC 20380-0001